

Secure User-Driven Pairing with Implantable Medical Devices

by

Mo Zhang

ORCID: 0000-0002-7302-9088

A thesis submitted in total fulfilment for the
jointly awarded degree of Doctor of Philosophy

at the
School of Computing and Information Systems
THE UNIVERSITY OF MELBOURNE
and the
School of Computer Science
UNIVERSITY OF BIRMINGHAM

December 2024

UNIVERSITY OF
BIRMINGHAM

University of Birmingham Research Archive

e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

Abstract

Implantable Medical Devices (IMDs), such as pacemakers and insulin pumps, are widely deployed today and often use wireless communication with external devices. Establishing a secure communication channel with an IMD can be life-critical. However, this is challenging in practice due to the special properties and needs of these devices. To address this issue, researchers and medical device companies have developed various types of pairing techniques for IMDs, aiming to bootstrap a cryptographic key between an IMD and an external device to secure the subsequent wireless communications.

In this thesis, we investigate these techniques and analyze their security, usability, and deployability (*i.e.*, whether the methods can be deployed in real-world devices) properties. Specifically, we highlight a worrying drawback in existing work: The pairing process with an IMD is always designed to be imperceptible to the patient. Assume a patient with a pacemaker is in a crowded public space (like a bus or a mall), a malicious adversary could potentially approach the patient and silently pair with their pacemaker without the patient noticing, which can lead to life-threatening consequences, *e.g.*, changing their therapy.

To mitigate this concern, we introduce several novel user-driven IMD pairing techniques, where pairing can be completed only after the user conducts specific physical interactions with the patient's body, *e.g.*, tapping on their body for a few times. Our approach makes the pairing process highly perceptible, and thus allows the patient to detect and stop the pairing at any time. We implement prototypes for our proposed techniques, and thoroughly evaluate them with human participants. Furthermore, we comprehensively validate whether our protocol can be potentially deployed in today's modern IMD products. The results demonstrate promising potential for our proposed pairing techniques. Our findings point out a new direction for future user-friendly design of security solutions for IMDs, and can provide value to both academia and medical device industry.

Declaration

This is to certify that

1. The thesis comprises only my original work towards the PhD.
2. Due acknowledgment has been made in the text to all other materials used.
3. Appropriate ethics procedures and guidelines have been followed to conduct this research.
4. The thesis follows the word length requirements.

Signed: Mo Zhang

Date: March 2024

Preface

The research presented in this thesis was conducted at the School of Computing and Information Systems, the University of Melbourne; and the School of Computer Science, University of Birmingham. This thesis comprises two under-review (in progress) articles and two peer-reviewed article, indicated by Roman numerals (articles I-IV). We include the main content of these articles in the context of this thesis. While several co-authors contributed to these articles, I declare that I am the primary author and have contributed the majority (greater than 50%) of the work in each of the following publications (more details of my roles and contributions are given in Section 1.3.1).

Article I

Mo Zhang, Eduard Marin, Mark Ryan, Vassilis Kostakos, Toby Murray, Benjamin Tag, and David Oswald. OOBKey: Key Exchange with Implantable Medical Devices Using Out-Of-Band Channels. Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES). 2024. URL <https://dl.acm.org/doi/10.1145/3664476.3670876>

Published by ACM on July 2024.

Article II

Mo Zhang, Chaofan Wang, Songyan Teng, David Oswald, Toby Murray, Eduard Marin, Bingyi Han, Ying Ma, and Vassilis Kostakos. Towards Secure User-Driven Pairing with Pacemakers.

In revision following peer review by the ACM Symposium on User Interface Software and Technology (UIST).

Ethics reference (University of Melbourne): 2022-24851

Ethics reference (University of Birmingham): 2022-0255

Article III

Mo Zhang, Chaofan Wang, Weiwei Jiang, David Oswald, Toby Murray, Eduard Marin, Jing Wei, Mark Ryan, and Vassilis Kostakos. Using Vibration for Secure Pairing with Implantable Medical Devices: Development and Usability Study.

In revision following peer review by the Journal of Medical Internet Research (JMIR) Biomedical Engineering.

Ethics reference (University of Melbourne): 2022-24851

Ethics reference (University of Birmingham): 2022-0255

Article IV

Mo Zhang, Eduard Marin, David Oswald, and Dave Singelée. FuzzyKey: Comparing Fuzzy Cryptographic Primitives on Resource-Constrained Devices. 20th Smart Card Research and Advanced Application Conference (CARDIS). 2021. URL https://doi.org/10.1007/978-3-030-97348-3_16

Published by Springer Nature on November 2021. We have obtained the copyright license (to use the materials in the thesis) from Springer Nature.

I would like to acknowledge the following sources of funding:

- Raymond Priestley Program Scholarship
(Organized by the University of Melbourne and University of Birmingham)

Article I and IV received minimal editing during the publication process, and the remaining thesis does not receive other third-party editorial assistance. No work incorporated in this thesis is completed outside my graduate researcher candidature. Ethical approval was not required for Article I and IV. The other studies presented in Articles II and III received ethical approval from the Human Ethics Advisory Group at both the University of Melbourne and University of Birmingham.

In this thesis, Chapter 2 (Background) includes a literature review encompassing relevant contents from all four articles, and Chapter 3 (Methodology) details the methods commonly used in the research presented in subsequent chapters. Furthermore, Chapter 5 is adapted from Articles I and II, Chapter 6 is adapted from Article III, and Chapter 7 is adapted from Article IV. All materials are appropriately integrated in the body of the thesis, and are adequately identified and referenced.

Acknowledgements

This thesis is the result of a long journey that would not have been possible without the help and support of many people and institutions.

It is a great honor to be a joint PhD candidate between the University of Melbourne and University of Birmingham. I want to thank these two amazing universities for organizing this great opportunity. The PhD journey across two countries and universities was absolutely fantastic. I also want to thank the Raymond Priestley Scholarship provided by these two universities, which allowed me to be fully dedicated to research without any concerns.

I want to give special thanks to my principal supervisors from the two universities: Prof. Vassilis Kostakos from Melbourne and Prof. David Oswald from Birmingham. I still clearly remember the first interview meetings we had in 2019 before I started the PhD. Thank you both so much for trusting me and giving me this opportunity that definitely changed my life. I also want to thank my co-supervisors: Prof. Toby Murray from Melbourne and Prof. Flavio Garcia from Birmingham. I cannot thank all of you enough for the help and support you have given over the years. I feel so pleased and honored to be your student. I also want to thank Prof. George Buchanan from Melbourne and Prof. Tom Chothia from Birmingham for chairing the PhD supervision teams who have always believed in me.

During my stay at the University of Melbourne and University of Birmingham, I have had the opportunity to write papers with very talented researchers. My special thanks go to all my co-authors: Vassilis Kostakos, David Oswald, Toby Murray, Eduard Marin, Mark Ryan, Dave Singelée, Chaofan Wang, Weiwei Jiang, Wei Jing, Songyan Teng, Benjamin Tag, and Ying Ma. Working with you was always a joy, and you have taught me so many things.

Working at both universities has been (and will always be) a great and unforgettable experience. I want to thank my friends in Birmingham: Xiaoyu Qin, Jing Zhang, Zitai Chen, Yuhang Qu, Huijuan Ren (Kaka), and others. I will never forget the days during the COVID-19 pandemic when we supported each other. I also thank my colleagues from the TEEFaulting team. In Melbourne, I want to thank my friends in the Ubi group: Wafa Johal, Jarrod Knibbe, Chaofan Wang, Wei Jing, Ying Ma, Kangning Yang, Weiwei Jiang, Ulan Kelesbekov, Songyan Teng, Brandon Syiem, Joe Brailsford, Saumya Pareek, Bingyi Han, Qiushi Zhou, Andrew Irlitti, and so many other people. You have been such warm friends, and I will always cherish the laughs we shared. I also want to give special thanks to some friends outside the Ubi group: Rui Liu, Shu Liu, Yilin Geng, Zhongyi

Bai (Marvin), Siqi Xia, Bingyi Han, Minbing Chen (Bryan), Rui Xing, Miao Li, Wei Gao (Rena), Tianyi Zhang (Tina), and others. I cannot help but smile whenever I recall the moments I spent with all of you.

In particular, I wish to thank Xiaoyu Qin, who encouraged me to pursue a PhD at the first beginning. Without you, I would not have even considered it. I wish to thank Eduard Marin, who has been such a supportive and reliable friend. I have learned so much from you, both in academia and life. I also wish to thank Rui Liu and Shu Liu, and I hope our special friendship as “three waves” lasts forever. I also want to thank my mentors during my research internship at Telstra, Xiangyue Meng (Sean), Yong Yuan, and Sathya Chandrasekharan, for guiding me to exceed myself at all times.

Last but not least, I would like to give my biggest thanks to my parents, Yunfeng and Baoan; my sweetest girlfriend (we have spent over nine years together!), Yaqi; and my girlfriend’s parents, Xiaobing and Gang. There are not enough words to express how much I love you. I want to thank you for your unconditional support and love over the years, and you know that everything achieved so far is as much mine as it is yours. This thesis is dedicated to you.

The PhD was not an easy journey—the repeated rejections from the research community, the disruptions due to the COVID-19 pandemic, and the challenges faced when conducting user studies, among others. Each one felt immense at the moment I experienced it. However, now I can genuinely smile and thank these setbacks for making me a better and stronger person. I am proud to say that during all these years of my PhD, I have never regretted embarking on this journey. If I could travel back in time, I would still choose to apply for this program.

Thank you!

Mo Zhang
Guangzhou, March 2024

Contents

Abstract	i
Declaration	ii
Preface	iii
Acknowledgements	v
List of Figures	x
List of Tables	xii
Abbreviations	xiii
1 Introduction	1
1.1 Motivation	1
1.1.1 Challenges for Pairing with IMDs	2
1.1.2 Previous IMD Pairing Techniques	3
1.1.3 Limitations of Prior Work	5
1.2 Research Questions	6
1.3 Contributions	7
1.3.1 Role of the Author	9
1.4 Thesis Structure	9
1.5 Impact of COVID-19 Pandemic	10
2 Background	11
2.1 Assumptions and Requirements	12
2.1.1 Adversary Model	12
2.1.2 Patient Model	13
2.1.3 Requirements for IMD Pairing Protocols	13
2.2 Related Work on Pairing with IMDs	15
2.2.1 Conventional Pairing Techniques	15
2.2.2 Out-Of-Band Channel-Based Pairing Techniques	18
2.3 Motion-Based Paring Techniques with IoT Devices	25
2.4 Fuzzy Cryptographic Primitives	26
2.4.1 Preliminaries	27

2.4.2	Overview of Fuzzy Primitives	28
3	Methodology	34
3.1	Framework of Our IMD Pairing Techniques	35
3.1.1	The Cryptographic Protocol	35
3.1.2	Security Analysis with PAKE	38
3.1.3	Potential Attack Vectors	40
3.2	User Study	41
3.2.1	Prototype Implementation	41
3.2.2	Participant Recruitment	44
3.3	Evaluation Metrics	44
3.3.1	Accuracy	45
3.3.2	Security	45
3.3.3	Usability	46
3.3.4	Time Efficiency	47
4	Case Study of a State-Of-The-Art Pacemaker	48
4.1	Hardware Analysis	49
4.2	Study of the Pairing Process	50
4.2.1	The Pairing Protocol	51
4.2.2	Patient Perception	51
4.3	Conclusion	52
5	Using Bodily Motions to Pair with IMDs	53
5.1	Motivation	53
5.2	Method	55
5.2.1	Pairing Techniques	55
5.2.2	Prototypes Using Rotation	57
5.2.3	Prototypes Using Tapping	58
5.2.4	Algorithm Development	63
5.2.5	Data Collection	63
5.3	Results	64
5.3.1	Participant Performance	64
5.3.2	Pairing Accuracy	66
5.3.3	Security Assessment	68
5.3.4	Efficiency and Usability Scores	71
5.3.5	Learning Effect in Tap Prototypes	71
5.3.6	User Perspectives	72
5.3.7	Insights From the Co-Design Workshop	73
5.4	Discussion and Conclusion	74
6	Using Vibration to Pair with IMDs	77
6.1	Motivation	78
6.2	Method	79
6.2.1	Pairing Technique	80
6.2.2	Obtaining a Shared Secret From Human	80
6.2.3	Signal Processing Workflow	82
6.2.4	Data Collection	84

6.3	Results	85
6.3.1	Shared Secrets	85
6.3.2	Pairing Accuracy	87
6.3.3	Security Assessment	88
6.3.4	User Perspectives	90
6.3.5	Optimal Configurations	91
6.4	Discussion and Conclusion	91
6.4.1	Comparison with Prior Vibration-Based Work	92
6.4.2	Potential Health Implications with Vibrations	94
7	Comparing Fuzzy Cryptographic Primitives on Resource Constrained Devices	95
7.1	Motivation	95
7.2	Method	96
7.2.1	Design Security and Parameter Selection	97
7.2.2	Implementation	100
7.3	Results	103
7.3.1	Hamming Metric Constructions	103
7.3.2	Set Metric Constructions	106
7.4	Discussion and Conclusion	107
8	Conclusions and Future Directions	109
8.1	Current IMD Pairing Techniques and Drawbacks	109
8.2	Developing “Seamful” IMD Pairing Techniques	111
8.3	Deployment in Real-World Healthcare Systems	113
8.4	Security Against Side-Channel Attacks	115
8.5	User Training and Ethical Considerations	115
8.6	Limitations and Future Directions	118
8.7	Conclusion	120

List of Figures

1.1	Pairing between IMD and external device.	2
2.1	Fuzzy commitment	30
2.2	Fuzzy vault	30
2.3	Code-offset construction of fuzzy extractor	31
2.4	Syndrome construction of fuzzy extractor	32
2.5	Improved Juels–Sudan fuzzy extractor	33
2.6	Pinsketch construction of fuzzy extractor	33
3.1	Our pairing protocol. ① is OOB key exchange, which involves measuring sensor data and applying a fuzzy extractor scheme (the syndrome construction, see Section 2.4 for details) [1] to remove noise from these measurements. <i>Bitstring</i> is the process of converting a set of integers into a bit string. “Syndrome” stands for the syndrome generation phase of an error correction code, while “StrongExtractor” refers to the application of a strong extractor (such as a secure hash function [2]) on the data. $\oplus$ is XOR operation. “Decode” denotes the decoding part of an error correction code. ② is PAKE (here we use Diffie-Hellman Encrypted Key Exchange (DH-EKE) [3] as a concrete example). g and p are public parameters. $h()$ is a secure hash function, and $\text{Encrypt}(key, msg)$ refers to the encryption of the message msg with key . Note that although we use fuzzy extractor and DH-EKE, many other fuzzy primitives or PAKE methods may also be applicable.	37
3.2	Experimental setup for Study II (Chapter 5).	42
3.3	Experimental setup for Study III (Chapter 6).	43
4.1	Medtronic Azure pacemaker (A) and its programmer device (B & C). . .	49
5.1	The smartphone user interfaces used in the study.	57
5.2	Workflow of enhanced prototype for taps.	59
5.3	Data collection in user study.	64
5.4	Results for rotation prototypes. (a) Distribution of smartphone rotation angles in baseline and enhanced prototypes. (b) Distribution of mismatch in rotation angles (<i>i.e.</i> , subtraction between a pair of angles for the pacemaker and the smartphone) in both prototypes. (c) The False Rejection Rate (FRR) versus the number of performed rotations, with all scenarios having zero False Acceptance Rate (FAR). The red circle indicates optimal configurations with $FAR = 0$ and $FRR < 5\%$, using minimum required rotations. (d) Distribution of the time taken to complete each rotation in both prototypes.	65

5.5	Results for tap prototypes. (a) Distribution of smartphone tap time-intervals in baseline and enhanced prototypes. (b) Distribution of mismatch in tap time-intervals (<i>i.e.</i> , subtraction between a pair of time intervals for the pacemaker and the smartphone) in both prototypes. (c) The FRR versus the number of performed taps, with all scenarios having zero FAR. The blue and orange lines indicate the baseline and enhanced prototypes, respectively. The solid lines represent scenarios where adversaries are knowledgeable about the users' tapping habits, while the dashed lines show cases where they are not. The red circles indicate optimal configurations with $\text{FAR} = 0$ and $\text{FRR} < 5\%$, using minimum required taps.	66
5.6	Learning effect in tap prototypes. Here we split the results from Figure 5.5c, depending on which prototype participants used first. The red squares indicate the minimum number of taps for accurate pairing using the baseline prototype against adversaries of various capabilities (blue lines).	72
6.1	Performance of swept vibration using our prototype. The peaks shared between devices are marked with red arrows.	81
6.2	Performance of constant vibration (50 Hz, 1 s per cycle) by our prototype.	82
6.3	An overview of our pairing technique.	83
6.4	Experimental setup.	85
6.5	Distribution of peak location and mismatch among 24 participants.	86
6.6	FRR <i>vs.</i> number of attachment motions under different vibration conditions ($\text{FAR} = 0$ in all cases). The red circles indicate configurations with $\text{FRR} < 5\%$.	87
7.1	Energy consumption of Hamming metric primitives.	105
7.2	Energy consumption of set metric primitives.	107

List of Tables

1.1	Mapping of contributions to our papers.	8
2.1	Evaluation of security and usability properties. Symbols indicate whether a solution fulfills (●) or does not fulfill (○) a certain evaluation criterion. ● means that the evaluation result is highly controversial.	21
2.2	Evaluation of deployability properties in terms of hardware compatibility (Dep 1), time efficiency (Dep 2), and resource consumption (Dep 3). We discuss consumer device compatibility (Dep 4) separately in the main text.	23
3.1	Our System Usability Scale (SUS) questionnaire.	46
3.2	Our interview questions for Study II.	47
3.3	Our interview questions for Study III.	47
4.1	Hardware capability of Medtronic Azure’s SoC [4].	49
5.1	Criteria of the NIST statistical test suite. Proportion indicates how many of the total 15 criteria fall into that category.	60
5.2	Evaluation of rotation and tap pairing techniques. IA stands for impersonation attack (see Section 3.1.3). The reported entropy and data throughput values have already taken into account entropy loss due to mismatch correction between devices.	68
5.3	NIST 800-22 statistical test results for taps.	69
5.4	NIST 800-90B statistical test results for taps.	70
6.1	Setup for previous vibration-based secret transmission. The application context refers to the intended receiver device.	78
6.2	NIST 800-22 statistical test results for attachment motions.	88
6.3	Entropy of each attachment motion (unit is bit).	89
6.4	NIST 800-90B statistical test results for attachment motions.	90
6.5	Summary of well-performing pairing configurations.	91
7.1	Effects of timing side-channel defenses on timing leakage (measured by Welch’s t-test) and energy consumption.	102
7.2	Evaluation of Hamming and set metric methods	104
7.3	Main building blocks of fuzzy primitives for TX and RX.	105
7.4	Fuzzy primitives with lowest total energy cost on the Hamming metric (Syndrome extractor) and set metric (Pinske extractor).	106
8.1	Comparison with previous work, in terms of accuracy (FAR, FRR) and time efficiency (throughput).	112

Abbreviations

AES	A dvanced E ncryption S tandard
BCH	B ose- C haudhuri- H ocquenghem
BLE	B luetooth L ow E nergy
CVE	C ommon V ulnerabilities E xposures
DH-EKE	D iffie- H ellman E ncrypted K ey E xchange
DoS	D enial-of- S ervice
ECC	E rror C orrection C ode
ECG	E lectrocardiogram
ERM	E ccentric R otating M ass
FAR	F alse A cceptance R ate
FDA	F ood and D rug A dministration
FFT	F ast F ourier T ransform
FRR	F alse R ejection R ate
ICD	I mplantable C ardioverter D efibrillator
IMD	I mplantable M edical D evice
IMU	I nertial M easurement U nit
IoT	I nternet-of- T hings
IPI	I nter P ulse I nterval
LRA	L inear R esonant A ctuator
LSB	L east S ignificant B it
LTK	L ong- T erm K ey
MITM	M an-in-the-middle
NIST	N ational I nstitute of S tandards and T echnology
OOB	O ut-Of- B and
OOK	O n-Off- K eying
PAKE	P assword- A uthenticated K ey E xchange
PKI	P ublic K ey I nfrastructure
PUF	P hysical U nclonable F unction
PWM	P ulse W idth M odulation

RF	R adio F requency
RNG	R andom N umber G enerator
RS	R eed- S olomon
RTT	R ound- T rip T ime
SoC	S ystem- o n- C hip
STK	S hort- T erm K ey
SUS	S ystem U sability S cale
TLS	T ransport L ayer S ecurity

Chapter 1

Introduction

1.1 Motivation

The number of patients with an Implantable Medical Device (IMD), such as a pacemaker, an Implantable Cardioverter Defibrillator (ICD), a neurostimulator, or an insulin pump, has been rapidly growing for decades. According to [5], the global active IMD market was USD 15.21 billion in 2017, and is expected to grow to USD 30.42 billion by 2025. Over the last decade, IMDs have evolved significantly both internally and externally. Currently, IMDs are computationally much more complex and interconnected than ever before. Meanwhile, there is an ongoing effort towards reducing their size as much as possible in order to increase the patients' comfort. For example, Micra AV2 and VR2¹, the most advanced pacemakers from Medtronic (a global leader in medical devices), are only about the size of a large vitamin capsule [7].

Modern IMDs typically rely on a wireless interface to communicate with external devices. In particular, we can distinguish between programmer devices through which doctors can wirelessly send commands to the IMD (e.g., to change the patient's therapy) and gather telemetry data, and monitoring devices—located in the patient's home—which periodically collect patients' medical data and send it to the hospital, thus enabling remote monitoring of the patient's condition when they are at home. Another emerging trend that holds the potential to revolutionize patient care is the fact that IMDs have started to support standard wireless technologies, which enable connections with consumer products [8]. For instance, Medtronic has incorporated Bluetooth Low Energy (BLE) into IMDs [9, 10], and developed a smartphone app that allows the patient to view some basic characteristics of their IMDs (e.g., remaining battery level) over a

¹Micra AV2 and VR2 were approved by U.S. Food and Drug Administration (FDA) in May 2023 [6].

BLE link [11]. Clearly, such wireless connectivity provides greater convenience to patients and doctors [12]. For example, a patient carrying an IMD may undertake medical examinations and receive personalized treatment without the need to leave their houses.

However, wireless channels also introduce new security threats. Adversaries may eavesdrop on the wireless channel remotely and obtain sensitive patient information [13], or even worse, send malicious commands to the IMD to alter its settings [14]. The consequences of these attacks can be severe, as these commands can allow adversaries to deliver (or disable) a therapy with the goal of causing serious injury or even death. Unfortunately, previous studies have demonstrated that many IMDs available in the market today severely lack effective security mechanisms [13–17], leaving a large group of patients vulnerable targets. Although no real-world attacks against an IMD have been reported yet, the threats definitely exist—the former vice president of the United States, Dick Cheney, asked his doctor to disable the wireless interface of his pacemaker to thwart possible assassination attempts [18].

To address these issues, it is necessary to first establish a secure wireless communication channel between an IMD and an external device before any sensitive data or commands are transmitted. This desires a secure *IMD pairing protocol* in order to establish a **shared cryptographic key** (like a 128-bit key for Advanced Encryption Standard (AES) encryption) between the IMD and the external device, which then becomes the shared basis for securing all subsequent communications. The thesis focuses on this topic, aiming to present novel pairing techniques that are essential for the security of IMDs and, accordingly, the safety of patients.

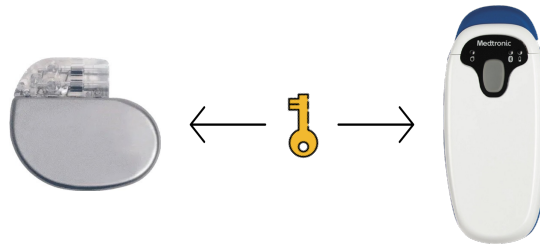


FIGURE 1.1: Pairing between IMD and external device.

1.1.1 Challenges for Pairing with IMDs

However, pairing in this context can be challenging due to the nature of current IMDs:

1. First, most of the IMDs have strict resource constraints, in particular limited memory and computational power. Furthermore, most of these devices are operated by a single non-rechargeable and non-replaceable battery which typically lasts between five and

- seven years (depending on the type of device and treatment) [19]. Once the battery is drained, the IMD is replaced through a surgical intervention that can pose risks to patients. Thus, in such resource-constrained devices every single joule matters. In Chapter 4, we will present a case study focusing on the resources of the Medtronic Azure pacemaker [8] as a representative example of a modern IMD product.
2. Second, the majority of IMDs are not physically accessible at all because they are implanted in the patient’s body. Hence, IMDs often do not have any input or output interfaces, such as a keypad or a screen. Wireless communication with the IMD is “invisible” to the patient, and there are no means to physically allow communication (e.g., by pressing a button).
 3. Third, most of the wireless connections that have to be made with these devices cannot rely on any prior trust relation. This is because these network connections are not static, *i.e.*, the set of external devices one needs to connect to can change quite often. For instance, during an *emergency* situation (e.g., heart disease patients that experience syncope symptoms and become unconscious [20]), the first doctor that is present (who may have never seen the patient before) might have to establish a communication link to the patient’s IMD. Equally, multiple devices such as a patient’s smartphone or a doctor’s tablet can vary during the IMD’s lifetime, considering that the patient may change their smartphone or go to another doctor.

It is noteworthy that the level of security required for pairing can vary based on the application scenario. For instance, printers that support Wi-Fi Direct often adopt a simplistic pairing method that does not require a password [21]. In contrast, life-critical IMDs are a special application that demands a high level of security, and thus the pairing process must be highly secure.

Due to these challenges and requirements, the process of pairing with IMDs is usually fundamentally different from pairing with traditional consumer devices, such as laptops and Internet-of-Things (IoT) devices (like a Bluetooth door lock).

1.1.2 Previous IMD Pairing Techniques

Several categories of IMD pairing techniques have been proposed, each with particular merits and limitations. We will briefly introduce them here, and provide a thorough review and analysis in Section 2.2.

The first category focuses on traditional symmetric key-based techniques. One approach is to embed a cryptographic key into the IMD during manufacturing [14, 22]. Any external device possessing this key can then directly pair with it. Although this method

is convenient and places minimal pressure on the IMD, it also introduces high risks, particularly in emergency situations where healthcare staff may not have access to the key. More importantly, if the key is compromised (e.g., exposed online), the IMD needs to be recalled, otherwise an adversary will be able to access the IMD indefinitely.

On the other hand, public key cryptography-based solutions, like those relying on digital certificates [23], are challenging to implement due to the need to establish a robust, global, and costly Public Key Infrastructure (PKI). Furthermore, this approach requires updates in the IMD whenever an external device is retired or compromised (e.g., stolen). However, IMDs typically lack an Internet connection to receive these updates.

There are also other designs to secure the wireless link between the IMD and the external device. One approach relies on distance-bounding protocol [24]. In this approach, both the external device and the IMD estimate their physical proximity as a criterion for pairing, which is often determined by calculating the Round-Trip Time (RTT) of an ultrasound signal—the time interval between sending the signal and receiving a response. This solution, however, requires an ultrasound transceiver in the IMD and precise measurements of the RTT, a demanding task for resource-constrained IMDs.

Moreover, proxy device-based protocols [25–27] have been proposed, where a proxy device carried around by the patient (such as a bracelet) serves as a security delegate for the IMD. This approach can work even with legacy IMDs as it does not need any modifications to them. However, it places the entire responsibility of security onto the patient, *i.e.*, requiring them to wear the proxy device constantly, which can lead to usability concerns, inconvenience, and security compromises in case the patient loses or misplaces the device.

Over the last decade, the use of Out-Of-Band (OOB) channels—any communication channel other than a wireless channel—has emerged as a promising way for pairing with IMDs. Pairings based on these channels usually offer special advantages over the conventional techniques. For example, they facilitate key exchanges on an ad-hoc basis, and do not rely on any prior secrets in the IMD.

One approach is to transmit the key via a low electrical current (e.g., 0.5 mA) from the skin to the IMD [28, 29] (or vice versa [16]), leveraging the human body’s conductivity. This method offers advantage that the current (*i.e.*, the secret key) is absorbed by the body, thus remaining concealed from adversaries [16, 29]. However, it poses safety concerns. For example, medical device companies have claimed that injecting a current near the IMD could potentially interfere with the device’s therapeutic functions [30].

Another strategy suggests using human physiological signals² as a shared randomness source to pair an IMD and an external device attached to the patient’s skin [32]. For example, previous work has proposed the use of Inter-Pulse-Interval (IPI)—the time interval between two consecutive R-peaks in an Electrocardiogram (ECG) signal—as a source of the key [33, 34]. However, the efficacy of known physiological signals as entropy sources is questionable [35]. In particular, they often lack sufficient randomness [36] and are susceptible to remote eavesdropping [37].

Previous work has also proposed to transmit the key through an audio [22] or vibration [38] signal. The primary advantage is that these signals are naturally perceptible to some extent, which allows patients to react accordingly if they notice a malicious pairing event. However, studies have indicated that these signals may be susceptible to eavesdropping from a short distance using a microphone [39]. The audio-based technique also requires a sound-producing unit in the IMD, which is not available in today’s IMD products. Additionally, previous research has raised concerns about the practical user perception of such methods due to their subdued signal amplitude [40, 41] and their deployability (such as the need for powerful sensors, as we will detail in Chapter 6).

1.1.3 Limitations of Prior Work

Overall, conventional cryptographic techniques (those based on symmetric/public keys) and other solutions are often unsuitable (or even not viable) in the context of IMDs. While the use of OOB channels carries unique advantages, it also entails trade-offs.

In this thesis, we specifically focus on one crucial aspect that has regularly been neglected: patient perception. Previous pairing techniques generally lack strong safeguards against adversaries in proximity of the patient, which try to pair with the IMD without the patient noticing. Most of the work attempts to make the IMD pairing as unobtrusive as possible to the patient, rendering the pairing process almost *imperceptible*. As we will demonstrate in Chapter 4, many advanced IMDs in the market [10, 11, 42] currently also follow this guideline.

Such a “seamless” design principle is admittedly prevalent in many everyday security systems, e.g., when surfing the Internet, users do not notice the pairing with a remote server using the Transport Layer Security (TLS) protocol. However, this heavily relies on the assumption that the pairing is sufficiently secure and is initiated by the legitimate user (both of which are true for TLS in the browser, but not necessarily for an IMD pairing). As this assumption is often not met [35, 37, 39, 43], we question if this guideline

²Note that physiological signals are random over time, unlike biometrics (e.g., fingerprint) that are person-specific and to a large extent invariant [31].

fits the IMD context, where the device is part of the patient and its security is life-critical. Instead, we argue for a shift towards a more “seamful” design, increasing security by enhancing patient perception during the pairing process.

1.2 Research Questions

Based on the background outlined in Section 1.1, we have identified three research questions addressed in this thesis:

RQ1: What are the existing IMD pairing methods used in academia and industry?

To address this research question, we carried out a comprehensive literature review of existing IMD pairing protocols (detailed in Chapter 2), and visited a government cardiology clinic to study and observe the pairing process of a popular pacemaker widely used worldwide (elaborated in Chapter 4).

Our study revealed a common drawback shared by both academic solutions and real-world medical devices: a lack of patient perception during the pairing process. We argue that this could be a significant design issue in this domain, which leads us to our next research question.

RQ2: Can the principle of “seamful” design be used to create new pairing protocols for IMDs?

In this thesis, we devise two main categories of pairing approaches with IMDs. The first approach leverages human bodily motions as an Out-of-Band channel for pairing, while the second approach utilizes vibration³—a signal inherently associated with a high level of perception—for IMD pairing. The details of these two techniques and their thorough evaluation with human participants are provided in Chapter 5 and Chapter 6, respectively.

Our results show promising potential for our proposed pairing techniques. However, as a reminder, IMDs are highly resource-constrained devices. It is essential to carefully assess whether a specific pairing method can be deployed in existing IMD products; otherwise, similar to a few previous proposals [22, 24], their practical utility may be questionable. This consideration leads to our third research question.

RQ3: How can our proposed protocols be implemented in such a way so that they can run on existing IMDs?

³Our proposed pairing technique is inherently different from previous vibration-based approaches and has special advantages, as shown in Chapter 6.

To answer this question, we design a lightweight cryptographic protocol for IMD pairing and meticulously ensure that the protocol can run on devices that simulate the capabilities of current IMDs (or even devices with lower capabilities). Specifically, we thoroughly benchmark a crucial cryptographic component in our pairing protocol, namely the fuzzy primitive, on resource-constrained devices (detailed in Chapter 7). Overall, we demonstrate that our proposed protocol can potentially be deployed in modern IMDs.

1.3 Contributions

The main contribution of this thesis is the development of secure, user-centric protocols for pairing an IMD with an external device, with a focus on maximizing the patient’s perception level of the pairing process to enhance security. We mainly contributed to this field in two ways: (i) by proposing novel cryptographic protocols that involve users to perform several simple actions, and (ii) by thoroughly examining these protocols’ practical usability among humans and their deployability in modern IMDs. The mapping of the contributions to our papers is shown in Table 1.1.

Following governing legislation, our research involving human subjects obtained approval from the ethics committees of the institutions associated with this work, including the University of Melbourne (where the user studies were conducted) and University of Birmingham. Our work mainly aims to propose new security protocols and countermeasures against certain attack vectors, and does not entail negative impacts on real-world IMDs and patients, such as identifying and disclosing the vulnerabilities of existing IMDs.

In detail, our contributions can be summarized as follows:

C1) Analyze a state-of-the-art wireless pacemaker. We conducted an analysis of the hardware capabilities of an advanced wireless pacemaker—Medtronic Azure [8]—as a representative example of modern IMDs. Additionally, we studied (but did not attack) the pairing protocol of this pacemaker model in a clinic setting. We pointed out potential design issues of this pacemaker’s pairing strategies, which are also shared by many other IMDs in the market and academic researches.

C2) Design novel user-driven pairing techniques for IMDs. To address the design shortcomings of prior IMD pairing solutions (in both clinical and academia settings), we devised two main categories of pairing techniques that allow a user (whether the patient or a doctor) to securely pair with the patient’s IMD by performing certain simple physical interactions. This process involves a careful design of the types of physical interactions required, as well as the construction of a cryptographic framework for the pairing, ensuring that it is secure, usable, and practically deployable.

C3) Examine the feasibility and usability of our proposed pairing techniques.

We developed three sets of prototypes that implemented our proposed pairing techniques. These prototypes closely matched the hardware capabilities of real-world IMDs, and were deployed in an environment that simulated the human body. We evaluated them in a laboratory setting including (i) a user study of 24 participants, and (ii) a co-design workshop with IMD domain experts. The results demonstrate promising potential of our pairing techniques.

C4) Study the deployability of our proposed pairing techniques. We investigated the deployability of implementing our pairing protocol in resource-constrained devices such as IMDs. Specifically focusing on fuzzy cryptographic primitives [19], a critical element in our pairing protocol (and many previous IMD pairing protocols), we summarized and compared the most widely used algorithms in the cybersecurity community. Furthermore, we provided open-source and constant-time (*i.e.*, the software always executes for a fixed time to defend against timing side-channel attacks [44]) implementations of these algorithms that can be directly deployed in many embedded systems with limited resources.

For clarity, throughout this thesis, we will refer to our studies that lead to the four contributions mentioned above as Study I, Study II, Study III, and Study IV. Detailed descriptions of these studies can be found in Chapter 4, Chapter 5, Chapter 6, and Chapter 7, respectively.

In keeping with the principle of openness, we publish all relevant data in this thesis, including the data from our user study and the software used in our prototypes, as open-source under a permissive license, following relevant ethics guidelines.

TABLE 1.1: Mapping of contributions to our papers.

Paper/Contribution	C1	C2	C3	C4
OOBKey: Key Exchange with Implantable Medical Devices Using Out-Of-Band Channels ⁴		✓	✓	
Towards Secure User-Driven Pairing with Pacemakers ⁵	✓	✓	✓	
Using Vibration for Secure Pairing with Implantable Medical Devices: Development and Usability Study ⁶		✓	✓	
FuzzyKey: Comparing Fuzzy Cryptographic Primitives on Resource-Constrained Devices [19]				✓

⁴In revision following peer review by the International Conference on Availability, Reliability and Security (ARES).

⁵In revision following peer review by the ACM Symposium on User Interface Software and Technology (UIST).

⁶In revision following peer review by the Journal of Medical Internet Research (JMIR) Biomedical Engineering.

1.3.1 Role of the Author

The research has resulted in four publications, which are incorporated into this thesis. These articles are published or under review in leading, international, and peer-reviewed conferences/journals in Ubiquitous Computing and Cybersecurity.

I was the first author of these publications and conducted the majority of the work (over 50%). Precisely, I introduced the ideas behind the work, initiated the research process, designed the experiments, and handled the prototype development, participant recruitment, user study, and data analysis. Furthermore, I prepared the manuscripts for submission and managed the revision process after receiving peer-review feedback. My co-authors provided me with feedback and advice on the study design and data analysis. Moreover, they contributed when preparing the manuscript of the publications. To reflect this contribution and show appreciation to my co-authors, the scientific term “we” is used throughout the main chapters of this thesis.

1.4 Thesis Structure

The remainder of this thesis is organized as follows.

Chapter 2 presents a comprehensive literature review to provide an overview of the context of pairing with IMDs, and to introduce related work conducted in this research area. This chapter also includes the common assumptions adopted by the IMD security research community, such as the adversary model and patient model.

Chapter 3 presents the theoretical framing and methodological decisions that are adopted by our research in this thesis. This chapter explains the motivations behind methodological decisions employed for data collection and data analysis.

Chapter 4 demonstrates a case study (Study I) we conducted to examine the pairing process of a real-world IMD in a clinical setting.

In Chapter 5 and Chapter 6, we detail two concrete IMD pairing techniques we propose to achieve secure, usable, and deployable pairings (Study II and III). This involves validating our techniques through the development of prototypes and evaluation by user studies, as well as presenting thorough data analysis procedures.

Chapter 7 presents our study evaluating and comparing the most commonly accepted fuzzy cryptographic primitives in resource-constrained embedded systems (Study IV), in order to ensure their deployability in modern IMDs and to select the best-performing ones.

Chapter 8 discusses the results and findings presented in this thesis in line with the research questions. Also, it details the considerations of potential side-channel attacks, required user training processes and ethical issues, and future directions for research on IMD pairing solutions. In the end, we conclude the thesis.

1.5 Impact of COVID-19 Pandemic

In accordance with the university's regulations related to thesis documentation, this section provides an honest and objective account of the impact of COVID-19 on my research. The pandemic has affected our work in two significant ways:

Firstly, Study I commenced approximately two months prior to the onset of the pandemic and continued for three months following its outbreak. However, due to the escalating situation, the study had to be prematurely terminated (further details are provided in Chapter 4). This unforeseen interruption not only changed my future research direction, but also resulted in the loss of nearly six months' work without any publications.

Secondly, my research targets the IMD patient demographic, and thus demands the recruitment of such individuals for our user studies in Study II and III. Unfortunately, the pandemic posed substantial risks to this vulnerable group, and rendered recruitment efforts incompatible with the university's ethics standards. Consequently, the majority of our study participants were not from the intended patient population.

Chapter 2

Background

This chapter provides an overview of the context of pairing with IMDs, and is structured into several sections:

In Section 2.1, we outline the core assumptions commonly adopted by the IMD security research community. This involves clearly delineating the capabilities of key stakeholders—namely, the adversary and the patient. Based on this, we further propose a set of requirements that an IMD pairing protocol should meet, in terms of security, usability, and deployability.

Section 2.2 presents a systematic review of previous proposals for pairing with an IMD. We rigorously evaluate them against the requirements we outlined above, with an emphasis on identifying the advantages and limitations of prior work. Note that this section only aims to summarize the proposals in academia, and the analysis of the pairing solutions used by real-world medical devices is given in Chapter 4.

To address the limitations of prior work, in Section 2.3, we explore insights from a different research area—pairing techniques with (non-implanted) IoT devices that involve user-performed motions. We analyze these approaches and highlight their benefits as well as potential applicability in the context of IMD pairing.

In Section 2.4, we focus on a critical cryptographic element in many previous IMD pairing protocols and the protocols we propose in this thesis: fuzzy cryptographic primitives. These algorithms are used to reconcile discrepancies in a shared secret between two devices without revealing the secret itself to the adversary. We summarize and discuss the most widely accepted algorithms within the cybersecurity community.

This chapter contains contents from all four articles encompassed in this thesis. In particular, Section 2.4 is based on the our published work in [19].

2.1 Assumptions and Requirements

In this section, the assumptions regarding the adversary and the patient, as well as the requirements for an IMD pairing protocol, are summarized based on our comprehensive review of relevant literature within the context of IMD security [14, 22, 24, 26, 27, 45, 46].

2.1.1 Adversary Model

We consider a sophisticated adversary following the Dolev-Yao model [47], who has full knowledge of the pairing protocol itself (but not the involved secrets) and the following capabilities:

Channel/network attacks: We assume that the adversary has full control of the wireless channel between the external device and the IMD. This implies that the adversary can either perform passive attacks by eavesdropping on the communication, or carry out active attacks, e.g., launch a Man-in-the-middle (MITM) attack by intercepting legitimate devices' signals and sending their own messages instead.

Furthermore, in cases when two devices pair by transmitting secrets through an OOB channel, the adversary may also be able to either eavesdrop or tamper with the data sent over that OOB channel, depending on the channel's characteristics (elaborated more in Section 2.2.2).

Hardware/software analysis: The adversary is able to gain access to examples of relevant hardware (e.g., by stealing a device programmer from hospital) and can profile and reverse-engineer them, obtaining all firmware and cryptographic keys (of that specific device). We however exclude the case that an adversary manipulates a target device (e.g., by replacing its firmware)—this issue can only be addressed through the use of secure hardware, which is orthogonal to the problem tackled in this thesis.

Proximity: The adversary is able to stay within close proximity (such as a few meters) to the patient, and may even try physical contact without the patient's awareness, e.g., when the patient is in a crowd or asleep.

The adversary may launch Denial-of-Service (DoS) attacks to manipulate the communication channel and disrupt the pairing. However, if pairing failures happen repeatedly, the external device can alert the user, who can take actions to investigate or report the suspected attacks. Furthermore, these attacks can be specifically mitigated using off-the-shelf solutions [22, 38].

Note that here we only present a general overview of the adversary’s capabilities. In the following chapters that focus on our concrete pairing protocols, we will expand upon this foundation to detail specific types of attacks an adversary might carry out, such as brute-force attacks and microphone based attacks.

2.1.2 Patient Model

Typically, patients with IMDs are elderly people. For instance, patients with cardiac IMDs, such as pacemaker and implantable cardioverter defibrillators, are usually around 63 to 75 years old [48]. There are two primary states for a patient:

Being conscious: This should be the most common state for a patient. Under this circumstance, we assume that the patient has fundamental cognitive and physical capabilities; otherwise, there are many more straightforward ways to cause severe impacts on the patient than attacking their IMDs.

Because the IMD and the patient are physically inseparable, it is reasonable to require the patient to take part of the responsibility for ensuring the IMD’s security: The patient should be aware of highly perceivable environment changes and potential suspicious activities. This could include, e.g., unusual signals indicative of IMD pairing attempts, such as continuous, loud sound or strong vibration that appears when the patient is not in hospital and under a routine examination. In such cases, the patient should react appropriately, e.g., by stopping the pairing, seeking help and escaping the scene.

We argue that the patient should not be responsible for anything else. For example, the patient does not need to wear a proxy device to protect themselves, or be able to detect a remote adversary who is eavesdropping on the communication channel.

Being unconscious: This can happen in specific extreme circumstances, in particular during emergencies. For instance, patients with cardiac implants may become unconscious when facing syncope symptoms [20]. Here, anyone with physical access to the patient should be allowed to pair with their IMD, because in this case there would be easier ways for adversaries to harm the patient than by hacking their IMD. The patient’s explicit consent is not required (as is the case for other emergency treatments), because their safety is the top priority.

2.1.3 Requirements for IMD Pairing Protocols

Based on the previously stated assumptions, we propose a set of core requirements one must consider in terms of security (*Sec*), usability (*Usa*), and deployability (*Dep*) for a pairing protocol in the IMD context:

(Sec 1) Channel-independent security. An adversary with full control of the wireless channel and some degree of control of the OOB channel must not be able to compromise the security of the protocol (e.g., must not be able to impersonate an IMD to the external device or vice versa). We will discuss the feasibility of compromising an OOB channel in the coming sections.

(Sec 2) Device-independent security. An adversary who can fully analyze the hardware/software involved in the pairing process (e.g., a device programmer bought online) must not be able to compromise the security of the protocol. In particular, this implies that all secrets are unique to a specific device and that there are no keys shared by multiple devices.

(Sec 3) Patient perception. For an adversary trying to pair with the patient’s IMD in close proximity, it is crucial that the patient is able to recognize this malicious activity, enabling them to seek assistance or escape the scene. Hence, the pairing process should be highly perceptible. In line with prior work [22, 38], we argue that the degree of perception should *exceed* mere subtle body contact.

It is worth noting that the feedback cue (if any) must be non-cancelable, *i.e.*, the pairing will not function without it. Otherwise, the adversary can easily remove it when performing attacks. For instance, additional visual or audio cues can certainly be integrated into the external device to enhance patient perception of the pairing process. However, these cues would not be essential to the underlying cryptographic protocol (IMDs do not have cameras or microphones to sense these signals). As such, an adversary can develop an external device that mutes all such cues, and thus may carry out an unauthorized pairing without the patient being aware.

(Usa 1) Simplicity. Patients and doctors should be able to easily understand and learn the steps of the pairing process.

(Usa 2) Executability. A patient or doctor should be able to execute the steps of the protocol. The protocol should consider physical restrictions the patient may have, e.g., some patients are unable to perform strenuous motions.

(Usa 3) Emergency availability. We define an emergency as a serious incident where the patient becomes unconscious and is unable to move, such as experiencing syncope symptoms in heart disease patients with cardiac implants [20]. In such cases, medical staff should be able to access the patient’s IMD without relying on Internet connection or other long-range communication links.

(Dep 1) Hardware compatibility. The pairing should be deployable to existing IMD products, and only utilizes the IMD’s existing hardware components. It can be difficult

to add additional hardware components just for pairing purposes, as such components will inevitably compete for space with the battery.

(Dep 2) Time efficiency. The protocol must be time-efficient. A long execution time (e.g., over half a minute) can negatively affect the user experience and emergency availability. This is especially if the protocol can fail sometimes due to environmental conditions, e.g., paramedic operational mistakes, which could further increase the total execution time.

(Dep 3) Resource consumption. The protocol must consume acceptable resources on the IMD in terms of hardware (e.g., extra memory) and energy consumption. To ensure this, one should carefully design and test the relevant program on the IMD to guarantee that the medical functions are not affected, and minimize both computational and communication overheads to conserve battery.

(Dep 4) Consumer device compatibility. As mentioned, nowadays the types of external devices are becoming increasingly diverse and are not limited to dedicated, proprietary devices built by medical organizations [11]. It would be advantageous for the pairing to be compatible with readily available consumer devices, *i.e.*, smartphones or tablets.

2.2 Related Work on Pairing with IMDs

In this section, we systematically review the previous IMD pairing approaches proposed in academia. As a reminder, the pairing aims to exchange a secret cryptographic key that is only known to the IMD and the external device, which is used to protect subsequent wireless communications.

We divide such work into two principal categories: (i) conventional pairing techniques, primarily grounded in the use of traditional cryptographic methods and wireless channels, and (ii) approaches that utilize an OOB channel for device pairing. We then offer a comprehensive evaluation of these prior work, aligning them with the unified set of requirements for IMD pairing protocols that we previously established in Section 2.1.3.

2.2.1 Conventional Pairing Techniques

Symmetric-key based. One approach to secure the connection between an IMD and an external device is by provisioning them with the same symmetric key during manufacturing [14, 22]. This method places little burden on the IMD. However, it is not compatible with the envisioned use of IMDs, where multiple personal devices, such as the patient's

smartphone or the doctor’s tablet, will need to be connected to the IMD during its life-time (e.g., when patients change their smartphone or decide to go to another doctor). Equally, in an emergency situation (e.g., if the patient becomes unconscious or has exacerbated symptoms) the doctor may need to get access to the patient’s IMD immediately using a new device programmer. The problem is that this device programmer may not be able to retrieve the correct symmetric key on time (e.g., due to not being able to identify the patient or not having a stable connection with the cloud). More importantly, if the symmetric key is compromised (e.g., exposed online), the IMD must be recalled to update a new key; otherwise, an adversary could access the IMD indefinitely.

Halperin et al. proposed a key diversification protocol to alleviate this issue [22]. In their protocol, an external device can derive the IMD’s pre-installed key using a key derivation function with a master key and the serial number of the IMD. In this way, secure communication can be established without requiring the external device to store a large number of symmetric keys or have access to the cloud. However, the master key is a single-point-of-failure, because it is shared by many external devices. All IMDs will be at stake if an adversary manages to obtain the master key from one of them (e.g., by stealing an external device and performing hardware attacks). Marin et al. proposed a solution to mitigate this problem by updating the master key periodically [14]. In this way, an adversary who obtains one master key can only derive valid symmetric keys of IMDs for a limited period of time. Nevertheless, the security of this method still depends on the assumption that the master key and the external devices are well protected, which is challenging to guarantee in practice.

Public-key cryptography based. One can also leverage public-key cryptography to realize pairing with IMDs. We provide a simple protocol as an example: the external device first transmits its public key to the IMD. Subsequently, the IMD generates a secret key, encrypts it using the received public key, and transmits the ciphertext to the external device. The security of the secret key is guaranteed because it can only be decrypted using the corresponding private key.

However, the biggest challenge of such a public key-based solution is that it is hard for the IMD to verify the legitimacy of public keys, *i.e.*, decide whether it belongs to a legitimate entity instead of a malicious adversary. One can address this issue using digital certificates, which are widely used in e.g., the TLS protocol. However, a digital certificate system is hard to manage as it requires the establishment of a worldwide robust and costly PKI. Moreover, because the external devices can be retired or compromised (e.g., stolen), the PKI must have proper mechanisms to revoke such illicit devices and update this information in the IMD. However, IMDs do not have an Internet connection to

receive these updates. Therefore, in practice, using public-key solutions also poses a number of challenges.

Distance-bounding protocols. These protocols are built around the principle that an external device—which is physically close enough to the IMD (e.g., a few centimeters)—is legitimate, while any device that exceeds this distance limit is malicious [24, 49]. At the core of such solutions is a method that leverages cryptography in combination with physics to allow both devices to reliably and securely estimate the distance to the other.

Distance measurements can be realized by calculating the RTT of a signal, *i.e.*, the time interval between sending a challenge and receiving the response. However, it can be very difficult to implement this on resource-constrained IMDs: the IMD must be able to accurately measure the RTT even if the signal travels very fast, e.g., near the speed of light for a typical wireless signal. Moreover, the processing time (including the time to process the challenge and compute the response) of the IMD must be very stable. Otherwise, the external device cannot tell if a long RTT is caused by a distance beyond range, or a software delay occurs in the IMD. Both criteria assume that the IMD is computationally relatively powerful, where in reality it is not.

Rasmussen et al. proposed a concrete solution that uses ultrasound to calculate the distance, relying on the fact that adversaries cannot transmit data on the audio channel using a signal which propagates faster than the speed of sound [24]. A significant advantage is that the speed of sound is relatively low, thus alleviating the computational burden of the IMD to some extent. However, this workaround opens the door for worm-hole attacks [50] and requires an ultrasound transceiver in the IMD, which may not be a viable option in practice.

Proxy device-based protocols. These protocols delegate security to a proxy device which can be carried around by the patient (e.g., a bracelet) [26, 27, 51]. One of their main advantages is that they do not require to modify the IMD, which makes them suitable for legacy devices with no (or limited) security. Unlike IMDs, the proxy device can have desirable hardware properties (e.g., higher computational power, rechargeable battery, Internet connection and input/output interfaces), which can be utilized to provide a high level of security, e.g., by using public-key cryptography between the proxy and an external device. After a secure connection is established between them, the proxy can distribute a session key to the IMD and the external device to build a secure channel. In emergency situations, a doctor can remove the proxy and connect with the IMD directly using an external device.

However, this solution requires that a secure communication between the IMD and the proxy can be established, e.g., based on pre-installed symmetric keys, which again can be

challenging in practice. Another drawback is that this method requires 24-hour wearing of the proxy device, which places the security burden on the patient, and also suffers in terms of usability and convenience. In addition, some concrete solutions [26, 51] protect the IMD by using the proxy to jam the wireless channel, which can be illegal in certain countries.

Summary. As we analyzed, conventional symmetric/public key-based pairing techniques and other solutions are often unsuitable (or even not viable) in the context of IMDs. Furthermore, some existing solutions can only be used in static contexts where the set of external devices the IMD can communicate with is very small and known at initialization time. Due to these severe limitations, we do not consider such methods in the rest of this thesis.

2.2.2 Out-Of-Band Channel-Based Pairing Techniques

In the past decade, the use of OOB channels (*i.e.*, any communication channels other than a wireless channel) has become a promising approach to establish security between an IMD and an external device.

The main intention to leverage OOB channels for IMD pairing is that they are generally considered to be more secure in contrast to long-range wireless channels, and thus are suitable for transmitting secret keys [16, 22, 38, 46]. Furthermore, compared to some conventional solutions (e.g., symmetric key-based ones), OOB channels do not rely on any prior shared information, which is especially advantageous in the context of IMDs. Moreover, the use of an OOB channel might also bring about additional benefits. For example, a vibration channel itself is perceptible to some extent, which might further enhance security as the user can sense when pairing is taking place.

However, compared to wireless channels, OOB channels typically have more limited bandwidth, can be more costly for data transmission and reception (e.g., in terms of the energy consumption), and may require additional hardware components (e.g., vibration transceivers). Therefore, the OOB channels *cannot* replace wireless channels and are used only for transporting a small amount of data, *i.e.*, a session key (or a secret value from which a session key can be generated). After a session key is agreed on, the parties can apply any standard encryption techniques and transmit the ciphertext through the wireless channel.

Tattoo or a bracelet as a visual channel. One possible way for establishing a key between an external device and an IMD—both in normal and emergency situations—is to tattoo the IMD key on the patient’s skin or to print it on a bracelet worn by the

patient [52]. This solution costs little and places little burden on the IMD, however, the applicability of this idea is debatable. For example, a tattoo can become unreadable after an accident, can disclose the patient’s condition to others, or be refused by patients due to the religious, aesthetic, or cultural concerns. Secondly, replacing the secret key on compromise is very difficult and requires tattoo removal. Equally, a bracelet can be damaged, lost, or stolen, which can lead to severe consequences.

Audio channel. Halperin et al. [22] were the first to propose transmitting a key through an audio channel. The main idea is that the IMD generates a random session key and broadcasts it as a modulated sound wave (4kHz signal generated by a piezo element embedded in the IMD) such that only an external device in close proximity (a few centimeters away) to the IMD can demodulate it correctly. One of the main advantages is that due to the nature of audio signals, patients may notice when a pairing process occurs.

The authors assume that a remote adversary cannot eavesdrop on the sound wave if the amplitude is carefully controlled. However, Halevi and Saxena [39] showed that adversaries can accurately capture the sound using an off-the-shelf microphone from several meters away. Furthermore, there is no evidence that a sound-generating module, like a piezo element, is available in current IMDs.

In contrast, Siddiqi et al. [53] proposed an IMD pairing scheme where the secret is transported using MHz-range ultrasound. The security of their solution is based on the assumption that the MHz ultrasound waves (generated by an IMD) can only be received by devices that touch the patient’s body. However, the authors only verified this security assumption in an acoustic software simulation but not in a real hardware set-up. Moreover, it is unclear if a MHz ultrasound transceiver can be miniaturized and embedded inside an IMD. Also, an ultrasound is not perceptible.

Vibration channel. Similar to the audio channel, researchers proposed to use vibration for pairing an IMD with an external device. The transmitter device (typically the external device) requires a vibration motor to generate the vibration signal, while the receiver (typically the IMD) uses an accelerometer to receive it. Saxena et al. [41] were the first to propose transmitting a key to an IMD via a vibration channel. Kim et al. [38] followed this direction and improved the time efficiency of the pairing process.

Similar to the audio channel solution, the activity of this channel is also to some extent perceivable by the patient. Crucially, unlike the audio channel, a strong advantage of this channel is that sending a vibration to an IMD requires the sender device to directly touch the patient’s skin. If an adversary was to overpower the signal with a very strong vibration, this could be easily noticed by the patient.

However, the vibration is essentially a low-frequency audio signal, which also inevitably emits acoustic side-channel signals [39]. Another limitation of existing vibration-based pairing approaches is their reliance on a powerful accelerometer in the IMD with a sampling rate of at least thousands of Hz, which may not be feasible for resource-constrained devices. More details of these previous work and their limitations will be discussed in Chapter 6.

Human body as a conductive channel. The human body can also be used for secret transmission due to its conductivity. A galvanic coupling technique was often proposed in this scenario. Concretely, the transmitter injects a low alternating current (e.g., 0.5 mA) into the skin/tissue, while the receiver can detect the voltage across two receiving electrodes elsewhere on the body. Wegmueller et al. [54] and Tomlinson et al. [29] proposed two implementations using galvanic coupling on an emulation of human tissue and skin, respectively. Marin et al. [16] also proposed a method where an electrical signal (secret key) is sent from a neurostimulator’s case to an external device that touches the human skin.

The common assumption of these solutions is that the current is absorbed by the tissue and emits little side-channel signals outside the body. Both [29] and [16] also tried to eavesdrop the emanations around the human body, but did not detect any. However, whether this method can totally defend against eavesdropping attacks is still debatable and requires further research.

Moreover, this method generally assumes that injecting a low-enough current is harmless for a patient carrying implants. However, this statement is not based on examinations conducted on real patients, but based on theoretical data for healthy people [54] or doctors’ experiences in medical practice [16]. In fact, medical device companies, such as Boston Scientific [30], have claimed that such current injection might interfere with the IMD’s therapy.

Physiological signal as a channel. In addition to the aforementioned solutions, where one of the devices is always responsible for generating and transmitting the secret to the other via an OOB channel, there are also solutions that rely on a shared entropy source like the patient’s body. Poon et al. [32] were the first to propose the usage of random physiological signals extracted from the patient’s body for establishing a cryptographic key between two devices. Physiological signals are random signals that vary over time, like the heartbeats. A standard approach consists of each device independently and synchronously taking a measurement of a chosen physiological signal. For an overview on the security and functional requirements that a physiological signal must satisfy to be used in a cryptographic protocol, we refer the reader to [35].

However, as shown by Marin et al. [35], the security of physiological signal-based cryptographic protocols is questionable, with attacks exploiting weaknesses or limitations in the selection of the physiological signal, the design of the protocols, or even their implementation. For example, Lin et al. [33] and Rostami et al. [34] proposed to use the IPI, *i.e.*, the time interval between two consecutive R-peaks of the ECG signal of the patient, for pairing purposes. However, Ortiz-Martin et al. [36] and Seepers et al. [37] have shown that IPI is not an appropriate selection. Overall, there is currently no known suitable physiological signal that meets all the requirements for being a qualified entropy source.

Notably, human biometrics, such as fingerprint and facial data, are not suitable for pairing with IMDs [19, 35]. This is because secure pairing requires that the exchanged key is valid only for a single pairing session, and is completely independent from previous and future keys, whereas biometrics are (to a large extent) permanent and unchangeable [31]. If an adversary obtains the patient’s biometric data, they could potentially use it to gain perpetual access to the IMD, which leads to severe consequences.

Analysis of previous OOB channels. We analyze previous OOB channel-based approaches against the requirements that we identified in Section 2.1.3. The results are summarized in Table 2.1 (for security and usability) and Table 2.2 (for deployability).

TABLE 2.1: Evaluation of security and usability properties. Symbols indicate whether a solution fulfills (●) or does not fulfill (○) a certain evaluation criterion. ● means that the evaluation result is highly controversial.

	Security			Usability		
	Channel-independent	Device-independent	Patient perception	Simplicity	Executability	Emergency availability
Visual channel [52]	○	●	○	●	●	●
Audio channel [22, 53]	○	●	●	●	●	●
Vibration channel [38, 41, 43]	○	●	●	●	●	●
Human body channel [16, 29, 54]	●	●	○	●	●	●
Physiological signal channel [33, 34]	○	●	○	●	●	●

Security. Previous attacks [35, 37, 39, 43] have exposed vulnerabilities of visual, audio¹, vibration and physiological signal-based OOB channels, to eavesdropping using microphones or cameras from several meters away, thus potentially violating *Sec 1*. To defend against these threats, certain countermeasures have been suggested. For instance, to

¹Although MHz-ultrasound channels are claimed to be confidential by [53], we argue that experimental studies are inadequate and that further testing is necessary.

safeguard audio and vibration channels, Kim et al. [38] and Anand and Saxena [43, 55] proposed using Gaussian white noise or masking signals to obscure the acoustic leaks. These approaches have shown promise in thwarting side-channel vulnerabilities against sophisticated eavesdropping techniques.

On the other hand, for *Sec 2*, all prior work does not require the protection of examples of relevant hardware/software to ensure security, because the exchanged secret is not shared by multiple devices, *i.e.*, it is either re-initialized in each pairing session or uniquely assigned to a specific device.

Of all the approaches, the audio (not ultrasound) and vibration channel-based solutions offer a degree of audible/tactile feedback to the patient. Nonetheless, the signal amplitudes were subdued in their initial designs [22, 38]. For instance, the vibration amplitude in [38] is only around 0.2g, which is confirmed to produce a limited level of perceptual intensity to humans [40]. Similarly, the effectiveness of the audio channel is also questioned by previous research [41]. While amplifying the signal might address this, it presents challenges as it could lead to more acoustic leakage (harming security) and potential discomfort for patients (harming usability). Furthermore, these techniques were not validated on real users. Thus, it is uncertain if they can genuinely satisfy *Sec 3* in practice.

Usability. The previous work commonly asserted the proposed techniques to have high usability (*i.e.*, simple to learn and easy to execute): The visual channel-based solution requires the user to retrieve the secret from a patient’s tattoo or personal item, while other solutions merely require the user to hold the external device near the patient or attach it to the patient’s body. Notably, such claims were seldom substantiated with user studies. Moreover, all methods are “plug-and-play” and the pairing process in emergency situations is identical to the one employed in hospitals.

Deployability. The deployability of a solution highly depends on the requirements and limitations of a specific application scenario. For example, a pacemaker is expected to have stricter size limitations than an insulin pump. Therefore, we do not make strict judgment on this criterion, but only provide the data from prior work as a reference in Table 2.2. Note that this data is not always complete, *e.g.*, only a few work provided an estimated energy consumption value. Additionally, the visual channel-based method is presented more conceptually, rather than giving a concrete implementation. Furthermore, it only requires a symmetric key to be pre-installed, thus placing little burden on the IMD. Therefore, we omit it in the following.

For *Dep 1*, all OOB channels require specific hardware components in the IMD: The audio-based solutions require a piezo element [22] or a MHz-ultrasound transceiver [53].

TABLE 2.2: Evaluation of deployability properties in terms of hardware compatibility (Dep 1), time efficiency (Dep 2), and resource consumption (Dep 3). We discuss consumer device compatibility (Dep 4) separately in the main text.

	Hardware Demands (IMD)	Time Efficiency	Energy Consumption (IMD)
Audio channel	piezo element [22]; MHz-ultrasound transceiver [53]	320 bps [22]; 50K bps (theoretical) [53]	claimed negligible [22]; < 15 mJ [53]
Vibration channel	accelerometer	1.5 bps (zero error rate mode) [41]; 20 bps [38]	< 4.5 mAh [38] (over IMD’s 90-month lifetime)
Human body	a pair of electrodes	85.3 bps [16]; 50K bps [29]; 4.8K bps [54]	≤ 2.75 mJ [29]
Physiological signal	biosensor (e.g., ECG sensor)	3.2 bps [33]; 4–5.3 bps [46, 56]	≤ 19.68 mJ [33]

Current IMDs lack these components, and no indications suggest their integration in the foreseeable future. The vibration channel demands an accelerometer within the IMD, a feature existing in cutting-edge IMDs. For example, the Medtronic pacemakers and ICDs already contain accelerometers for medical purposes² [57–59]. However, it is important to note that previous vibration-based protocols often require a powerful accelerometer in the IMD to reliably receive the vibration signal. For example, [38] utilizes an accelerometer with a sampling rate of 3200 Hz.

In contrast, the technique based on current transmission through the body needs a pair of electrodes/wires in the IMD to receive the current. To our knowledge, there is no evidence of their existence in IMDs (though their integration may be feasible). Moreover, physiological signal-based solutions leverage a biosensor, which can be inherently present in IMDs for medical use, e.g., the ECG sensor in cardiac implants.

We use bit rate (*i.e.*, bit per second) to represent time efficiency of each method. We observe that solutions based on audio channels and human body present high data rates, while the vibration and physiological signal-based methods yield considerably lower speed. If a 128-bit cryptographic key is to be exchanged, these methods may result in a pairing duration of dozens of seconds, which potentially violate *Dep 2*.

Among the energy consumption values in prior work, the physiological signal-based method reports the most energy with 19.68 mJ per execution. To evaluate this, we first estimate an IMD’s battery capacity using advanced pacemakers as an example: The energy density of a pacemaker’s Li/I₂ battery can reach 210 Wh/kg [60]. Hence, assuming the battery constitutes 80% of the total weight of the pacemaker, a 20 g pacemaker battery—approximated using the Medtronic Azure pacemaker [8] as an example—would have an energy capacity of roughly 4 Wh (14400 J). Therefore, the energy consumption of prior work should generally satisfy *Dep 3*. This is a rough estimation only, and we emphasize that it is vital to test the energy consumption carefully on the resource-constrained IMDs, as what we do in Chapter 7.

²An accelerometer is used to enable Medtronic’s Rate Response feature, *i.e.*, adapt the pacemaker’s pacing rate to changes in the patient’s physical activity.

For *Dep 4*, for the visual, audio, and vibration channels, commercial devices like smart-phones suffice as external devices, given their inherent features like keyboard, microphone, and vibration motor. However, human body channel is incompatible as it demands the external device to emit a current. The compatibility of physiological signal-based solutions depends on the type of physiological signal.

Limitations of existing work. As we have discussed so far, existing OOB channels offer features well-suited to the IMD pairing context; however, they also exhibit certain limitations:

Vulnerable to injection attacks. Current OOB channels generally lack strong safeguards against adversaries in proximity of the patient, who try to pair with the IMD (by actively injecting a key with it) without the patient noticing. This is a crucial aspect that has largely been overlooked by the research community: most prior works follow the design guideline that makes pairing with IMDs as unobtrusive as possible to the patient. The pairing process is invisible in most cases. We will show in Chapter 4 that many advanced IMDs in the market today [10, 11, 42] also follow this design principle.

However, this principle heavily relies on the assumption that the pairing is adequately secure and that the implications of a security breach are minor. We question if this guideline fits the IMD context, where the implant is essentially a part of the patient and its security is vital. For example, the convenience afforded by unobtrusiveness could potentially give an adversary the ability to perform attacks silently.

We emphasize that this vulnerability is especially evident in the public spaces of everyday life. For example, on a crowded bus, it may be difficult for the patient to recognize if a nearby person is trying to pair with their IMD. In such scenarios, proximity to the patient, or even brief physical contact with them (a necessary step in many previous IMD key exchange techniques [28, 29, 33, 38, 46]), could potentially be overlooked by the patient.

Vulnerable to eavesdropping attacks. OOB channels also face threats from sophisticated side-channel eavesdropping using microphones and cameras. Fortunately, over the last decade, such threats have been largely mitigated with various defense strategies [38, 43, 55, 61]. However, we identify another issue in previous works: they commonly propose using an OOB channel to *directly* exchange a session key. This simplifies an attack, as the adversary does not have to recover the secret in real-time during its transmission over the OOB channel. Instead, they can record the traffic and launch offline brute-force analysis, potentially increasing the likelihood to compromise the security [39].

Deployability issues. Many proposals cannot be deployed on existing modern IMD products, and are also not compatible with off-the-shelf consumer devices as external devices.

Another concern is the potential for a prolonged pairing time due to the relatively low data rates for some OOB channels. For example, vibration and physiological signal based channels can take dozens of seconds to exchange a 128-bit key, which poses usability and safety concerns.

2.3 Motion-Based Pairing Techniques with IoT Devices

In order to mitigate the injection attacks mentioned above, we draw inspiration from previous studies, which proposed to pair with (non-implanted) IoT devices by leveraging motions that users carry out. We review and analyze such work in this section.

One typical approach involves the initiator device (with an output interface such as a screen) generating a secret. This secret is mapped into a series of motions and displayed on the output interface. During the pairing process, the user is guided to interact with the IoT device and complete the motions. For instance, previous work realized pairing between an IoT and a laptop by having the user pat [62] or wave [63] the IoT device in certain patterns, following instructions displayed on the laptop's screen.

Different from the step-by-step guidance approach, previous work also pursued another direction where user is given more autonomy: users repeat a certain type of motion a few times, serving as a source of entropy. The underlying principle is that human behavior inherently exhibits a degree of randomness [64]. Under this approach, one work suggested pairing two IoT devices by holding them together in hand and shaking them [65]. Another approach facilitated pairing between two wearables by asking the user to walk for dozens of steps [66, 67]. In these scenarios, the secrets originate from the user's unique and (to some extent) random shaking gestures and gait patterns, respectively. Similarly, [68] introduced a pairing method between an IoT device (equipped with e.g., a button) and a smartphone. This method relied on a sequence of user-generated events, such as random button presses. Moreover, when pairing two IoTs with tangible screens, [69] proposed a technique based on simultaneous drawings on both screens.

However, the above methods cannot be directly transferred to IMDs. Some of them require direct physical contact with the device, which is impossible for IMDs implanted within the body [62, 68, 69]. Others cannot cope with situations where the patient has limited or no mobility, e.g., in case of disability or unconsciousness during emergencies. In these scenarios, the patient is unable to perform the required motions, such as moving their body to draw specific patterns [63], walking for dozens of steps [66, 67], or shaking the device by jumping [65]. On the other hand, it is difficult for medical staff to assist patients in performing these motions.

Nevertheless, these methods have provided us with valuable insights due to their unique advantages: (i) They provide users with an increased sense of control over their devices. While the user may not understand the underlying technology, they can feel active engagement in the pairing. (ii) The physical motions interacting with the target device are indispensable. Since IMDs are located inside the patient’s body, any physical interactions must involve the patient directly. This implies that the pairing can be designed highly perceptible. Consequently, patients hold the authority to decide whether to execute the pairing (unless being unconscious in an emergency), acting as an additional robust countermeasure against unauthorized access. This way, an adversary may not succeed merely by being in close proximity.

Another promising aspect of human motions is that they can be measured using inertial sensors, e.g., accelerometer and gyroscope [70]. Nowadays, inertial sensors are becoming more miniaturized and energy efficient, which makes them appropriate for advanced IMDs. As said, Medtronic has incorporated these sensors into their cardiac IMD products for medical purposes [57–59]. Moreover, inertial sensors might gain broader acceptance in IMDs in the foreseeable future. For example, many researchers have proposed to add these sensors on other types of IMDs for medical use, e.g., on insulin pumps as an adjunct to hypoglycemia treatment [71, 72]. Furthermore, inertial sensors have been prevalent on consumer devices like smartphones.

However, as acknowledged by many previous studies [63, 68, 73], using human physical interaction as a means for pairing can inevitably lead to potential visual and/or auditory side-channel leakage. We will discuss these risks in Section 8.4.

2.4 Fuzzy Cryptographic Primitives

The pairing process involves the IMD and the external device exchanging a secret, which is used to secure subsequent wireless communications. As mentioned above, previous work has adopted two primary ways for exchanging this secret: (i) one device transmits the secret to the other, or (ii) both devices simultaneously measure a shared entropy source. However, regardless of the method used, discrepancies in the exchanged secret can occur due to inherent noise, such as sensor measurement noise. For the pairing to be successful, it is crucial that both devices possess an identical secret.

This section reviews a specific cryptographic component, namely the fuzzy (cryptographic) primitive, which is designed to rectify such discrepancies without revealing the secret itself to the adversary and compromising the security level. Fuzzy primitives

play a crucial role in many existing IMD pairing protocols, and are also adopted by our proposed protocols in this thesis.

This section is based on the our published work in [19].

2.4.1 Preliminaries

We begin by presenting the relevant background needed to understand fuzzy primitives. Elementary computations for fuzzy primitives are in $GF(2^m)$, which contains 2^m distinct elements represented as $\{0, 1, \dots, 2^m - 1\}$. Addition (or subtraction) of two elements is equivalent to the $\oplus$ (XOR) operation. Multiplication in $GF(2^m)$ is polynomial multiplication modulo an irreducible polynomial of degree m . Equivalently, multiplication can be efficiently realized with log/antilog tables for resource-constrained devices [74].

A **metric space** M is a finite set. For each M , there is a definite integer distance $dist(m_1, m_2)$ between any two elements m_1 and m_2 . The fuzzy primitives discussed in this thesis rely on two different kinds of metric spaces: (i) *Hamming metric space* and (ii) *set metric space*. In a Hamming metric space, $M = F^\ell$ for an alphabet F . $dist(m_1, m_2)$ in M is the Hamming distance, which is the number of positions that m_1 differs from m_2 . For example, for $M = \{0, 1\}^3$, $dist(\{0, 0, 0\}, \{1, 1, 1\}) = 3$. Besides, the number of non-zero elements in m_1 is called m_1 's Hamming weight. In a set metric space, M contains all s -element subsets of a universe U . $dist()$ in M is the set difference, which is the size of symmetric difference (defined by $symdiff(m_1, m_2) = \{x \in m_1 \cup m_2 \mid x \notin m_1 \cap m_2\}$). For example, $U = GF(2^3)$ and $s = 3$, $dist(\{0, 1, 2\}, \{0, 1, 3\}) = 2$. $dist(m_1, m_2)$ is an even number when the size of m_1 and m_2 is the same.

The data type of fuzzy primitives' input—the (potentially mismatched) secrets possessed by the IMD and the external device—working on these two metrics is different. The Hamming metric operates on bit strings, whereas the set metric works with a set of elements. The choice between these metrics is mainly guided by the nature of the use case, e.g., whether the exchanged secret is more readily converted into a bit string or a set.

For instance, when utilizing patient's ECG data as a shared entropy source for pairing (see Section 2.2.2), Hamming metric methods can be used, where a bit string can be created by concatenating the bit representations of each piece of IPI from the ECG measurement. However, because a bit string is consecutive, these methods are sensitive to dislocation and erasure errors on the measurements (e.g., due to peak misdetection of heart beats [75]). One bit erasure at the start of a bit string might lead to a significant increase in the Hamming distance. Set metric methods alleviate these problems to a

certain extent. The input in this case is a set, and even if there are order-difference or erasure problems on set elements, the set difference will not vary substantially. However, converting, e.g., the ECG data into a set whose elements are uniformly distributed in U can be complicated, especially when the size of U is large.

Error Correction Codes (ECCs) are frequently used to achieve error-tolerance in fuzzy primitives. An ECC comprises *encoding* and *decoding* phases. In the encoding phase, the original data is encoded as a codeword, where some form of redundancy is added. When errors appear in the codeword, the decoding phase recovers the original data if the total number of errors is below the error tolerance limit. ECCs are represented by the triple $\{n, k, t\}$, where n is the number of symbols of the codeword, k is the number of symbols of the data ($k < n$), and t is the maximum number of errors that can be corrected. The error tolerance is then t/n .

For linear ECC, the syndrome can be generally calculated as $c \cdot H^T$, where c (the *codeword*) is a row vector and H is the parity check matrix. This syndrome contains all information necessary for decoding. If a codeword c has no errors, its syndrome $\text{syn}(c)$ is an all-zero vector. If errors appear, the syndrome of a malformed codeword c' is the syndrome of the error vector e because $\text{syn}(c') = \text{syn}(c) + \text{syn}(e) = 0 + \text{syn}(e) = \text{syn}(e)$. By decoding $\text{syn}(e)$, we obtain e if its Hamming weight (*i.e.*, the number of errors) is $\leq t$.

In this thesis, we focus on two common linear ECCs, namely Reed-Solomon (RS) and Bose-Chaudhuri-Hocquenghem (BCH) codes. Note that all BCH codes we use are binary codes. RS and BCH were recommended in the first papers on fuzzy primitives [1, 76]. They not only provide flexible parameter selection, but have efficient encoding and decoding methods. As we will show in Chapter 7, an efficient ECC can greatly improve the performance of the fuzzy cryptographic algorithms.

2.4.2 Overview of Fuzzy Primitives

Below, we provide an overview of the most widely accepted fuzzy cryptographic primitives in the research community. We distinguish between two classes of constructions: (i) the ones based on Hamming distance and (ii) those that rely on set difference. Hamming metric constructions include the *fuzzy commitment* [76], *code-offset* [1] and *syndrome* [1], whereas the *fuzzy vault* [77], *improved Juels–Sudan* [1] and *Pinsketch* [1] are set metric constructions.

To better understand and compare different fuzzy primitives, we first define a common goal for them: they aim to enable the IMD and the external device to agree on a 128-bit

cryptographic *key* (which is sufficient for, e.g., AES encryption) based on a previously *shared secret* (which is likely to be not identical but rather similar between two devices). We denote the transmitter and receiver devices as TX and RX, respectively. Both fuzzy commitment and fuzzy vault are used to securely transport a cryptographic key from TX to RX that is concealed by the shared secret. In contrast, all constructions based on fuzzy extractors extract the key from the shared secret itself³.

We refer to the secrets possessed by TX and RX as sc and sc' , respectively. $\xleftarrow{Ext}$ denotes the extraction of sc or sc' , including signal sampling and analog-to-digital conversion. $\xleftarrow{R}$ denotes random value generation. $\xleftarrow{Shuffle}$ refers to randomly mixing elements in a set, while the process of calculating the roots of a polynomial is denoted as $\xleftarrow{roots}$. $(0,1)^\ell$ represents an ℓ -bit length string. We use $\{x,y\}^s$ to represent a set consisting of s distinct elements. In all fuzzy cryptographic constructions described below, the first step is for TX and RX to extract sc and sc' , respectively. In the rest of this section, we will omit this step when describing each of the fuzzy cryptographic constructions.

1) Fuzzy commitment. Figure 2.1 illustrates how the fuzzy commitment scheme can be used to securely transport a cryptographic key from TX to RX. Initially, TX generates a random *key* and encodes it to form a *codeword*. Subsequently, TX masks the *codeword* by XORing it with sc and then sends the resulting value (denoted by fc) to RX. Upon receiving fc , RX generates $codeword'$ by XORing fc with sc' . Only if the distance between *codeword* and *codeword'* is less than the maximum number of errors that the ECC can correct, RX can successfully recover the cryptographic key previously generated by TX.

Much like standard commitment schemes, fuzzy commitment schemes are both *concealing* and *binding*. The former indicates that an adversary is unable to open the commitment in order to recover the committed value, whereas the latter refers to the fact that it is impossible to modify the value after committing to it. The binding process can e.g., be realized by TX sending the hash of the *codeword* with fc to RX. We omit this step and focus only on the key exchange step, similar to [68]. However, it is worth noting that the binding process might be necessary in more complex cryptographic protocols.

2) Fuzzy vault. The fuzzy vault, shown in Figure 2.2, is an order-invariant version of the fuzzy commitment. It is designed to “lock” a cryptographic key into a vault using a set of features A , such that the vault can be unlocked only by using a set of features B that is sufficiently similar to A (under some similarity metric). Concretely, the fuzzy vault works as follows: initially, TX generates a cryptographic key and then embeds it as

³Dodis et al. also defined several fuzzy extractor constructions on the *edit metric* [1]. However, these constructions are not suitable for the use case of IMD pairing. Therefore, we do not consider them in the thesis.

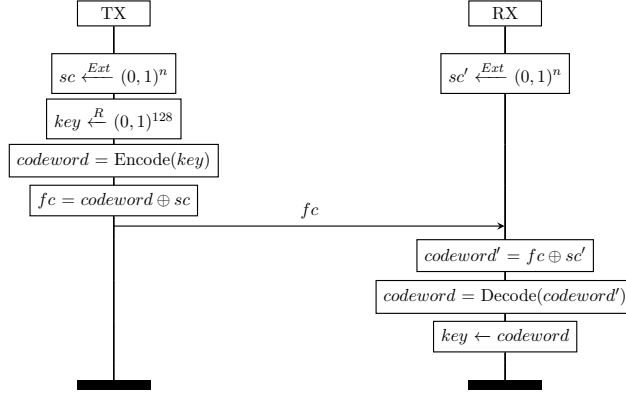


FIGURE 2.1: Fuzzy commitment

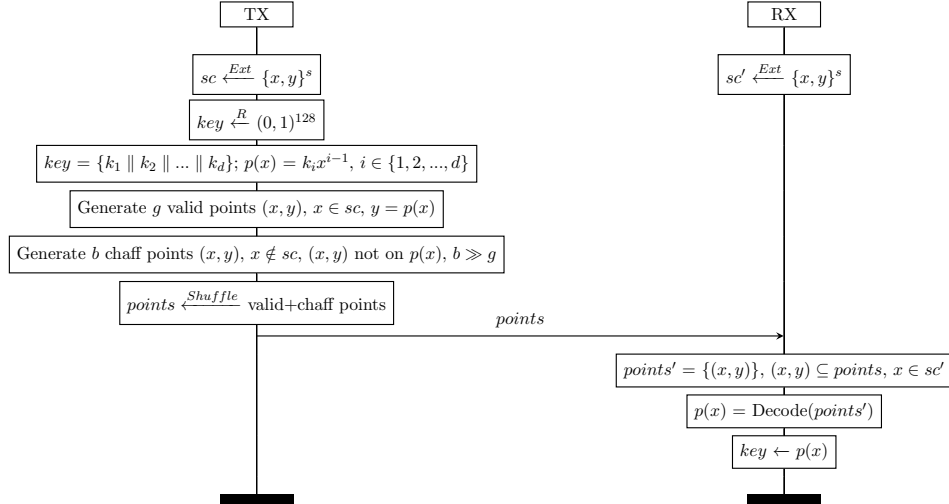


FIGURE 2.2: Fuzzy vault

coefficients in a univariate polynomial $p()$. Afterwards, TX generates a number of valid points (x, y) , where x is an element in the set sc and $y = p(x)$, as well as invalid points (also known as ‘chaff points’) that do not lie on $p()$ used for concealing the valid points. Both the valid and chaff points are then mixed and sent to RX. For each of the received points (x, y) , RX verifies whether x is in sc' , and if so, keeps the point. Once this is completed, RX performs polynomial reconstruction based on all the kept points. Only if there is a sufficiently large overlap between sc and sc' , RX can successfully reconstruct $p()$ and recover the key by concatenating the coefficients of the polynomial.

3) Fuzzy extractors. In general, all fuzzy extractors comprise two main steps. First, the mismatches of the shared secret at the TX and RX sides are corrected by sharing

so-called helper data. Afterwards, both sides use the agreed identical secret as the input to a strong random extractor (e.g., a secure hash function), whose output is the cryptographic key shared between TX and RX. In this thesis, we consider four concrete constructions based on fuzzy extractors in Hamming and set metrics.

3.1) Code-offset construction. This construction, shown in Figure 2.3, is similar to the fuzzy commitment scheme. The main difference is that sc is the secret in this construction, while in the fuzzy commitment, sc is used to conceal the key. In particular, the code-offset construction works as follows: first, TX generates a random *nonce* and encodes it as *codeword* using a pre-defined $\{n, k, t\}$ ECC. Then, TX calculates $ss = \text{codeword} \oplus sc$ and sends it to RX, which in turn recovers the $\text{codeword}' = ss \oplus sc' = \text{codeword} \oplus sc \oplus sc'$. Note that $\text{codeword}'$ and codeword can be different due to mismatches between sc and sc' . Yet, these errors can be corrected in the ECC decoding if the mismatch rate is under the error tolerance of the ECC. In this case, RX obtains $sc = \text{Decode}(\text{codeword}') \oplus ss$. Finally, both TX and RX utilize a strong extractor to extract the same key from the agreed sc .

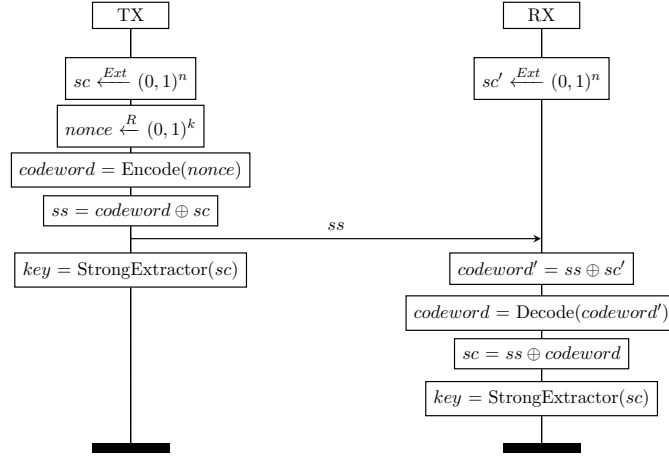


FIGURE 2.3: Code-offset construction of fuzzy extractor

3.2) Syndrome construction. This construction, shown in Figure 2.4, is based on the syndrome decoding of an ECC. First, both TX and RX regard sc and sc' as a *codeword* and calculate their syndromes as syn and syn' , respectively. The syndrome calculation method depends on the underlying ECC. Then, TX sends its syndrome syn to RX. After receiving syn , RX calculates $syn \oplus syn'$, which would result in an all-zero vector if $sc = sc'$. Otherwise, assuming that the mismatch vector is $mis = sc \oplus sc'$, $syn \oplus syn'$ is the syndrome of mis , whose decoding result is mis if the mismatch rate is within bounds. Then, one can recover $sc = mis \oplus sc'$. Finally, both TX and

RX utilize a strong extractor to extract the same key from the agreed sc . Compared to the code-offset construction, an advantage is that the length of the syndrome is always smaller than that of the codeword, substantially reducing the communication overhead.

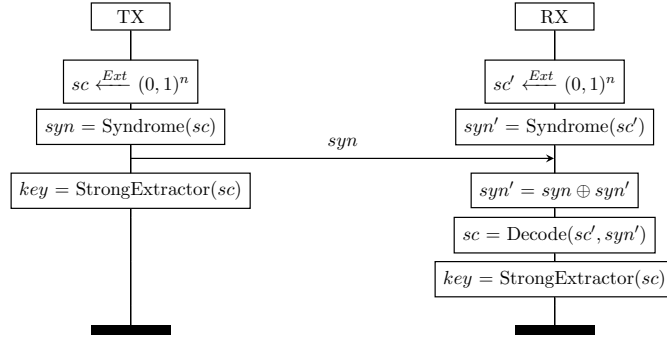


FIGURE 2.4: Syndrome construction of fuzzy extractor

3.3) Improved Juels–Sudan construction. Shown in Figure 2.5, this construction relies on polynomial reconstruction. First, TX assumes that there is a monic polynomial $p(x) = \prod_{w \in sc} (x - w)$, whose roots are the elements in sc . This polynomial is regarded as the sum of two parts $p_{high}() + p_{low}()$, where $p_{high}()$ consists of the first $t + 1$ elements of it and $p_{low}()$ comprises the rest of the elements. TX calculates the coefficients of $p_{high}()$ and sends its lowest t coefficients (the first coefficient is 1) to RX. Then, RX generates several points (x, y) , where x is in sc' and $y = p_{high}(x)$. If $sc' \approx sc$, most of the generated points will also be on $p_{low}()$, so we can reconstruct $p_{low}()$ successfully. Afterwards, RX can obtain sc by searching for all the roots of $p() = p_{high}() + p_{low}()$. Finally, both TX and RX utilize a strong extractor to extract the same key from the agreed sc . Compared with the fuzzy vault, this construction has much lower communication overhead because only some polynomial coefficients have to be sent.

3.4) Pins sketch construction. Figure 2.6 shows the “Pins sketch” construction for fuzzy extractors. This is based on an ECC. The main idea of this method is as follows: assuming the universe size is u (u is restricted to be $2^m - 1$ for some integer m), a set set can be viewed as a vector $\{0, 1\}^u$, with 1 at position where $x \in set$ and 0 otherwise. In this way, we regard sc and sc' as two u -length vectors v, v' whose Hamming weight is the set size s . First, both TX and RX calculate the syndromes of v and v' as $sstx$ and $ssrx$, respectively. Then, TX sends its syndrome $sstx$ to RX, while RX calculates $syn = sstx \oplus ssrx$. If the mismatch rate is under the error tolerance of the underlying

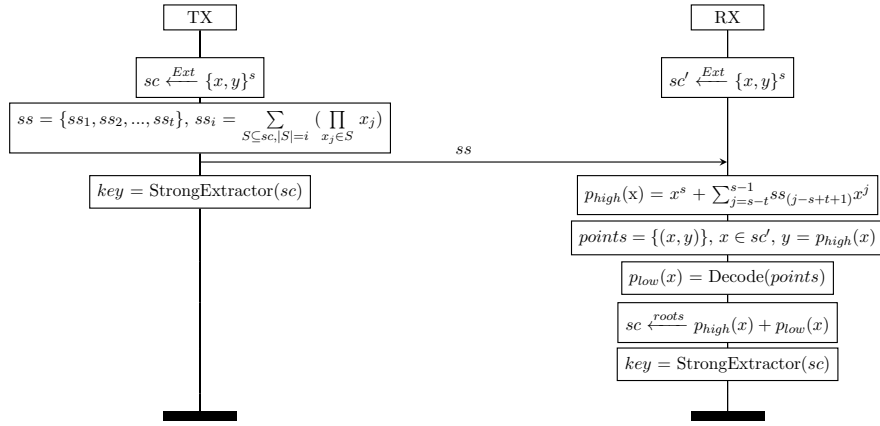


FIGURE 2.5: Improved Juels–Sudan fuzzy extractor

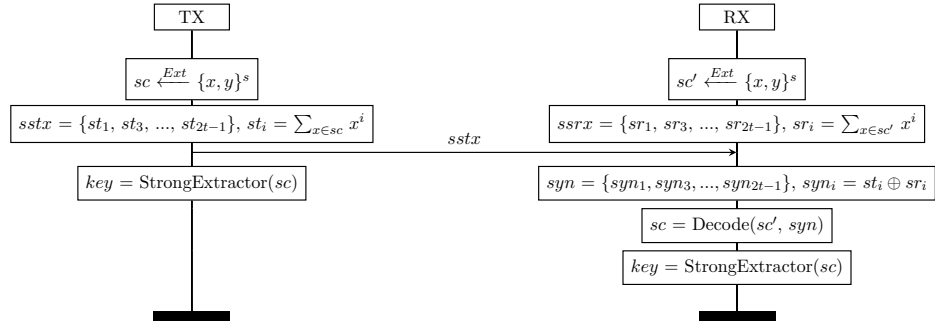


FIGURE 2.6: Pinskech construction of fuzzy extractor

ECC, the syndrome decoding result of syn is the symmetric difference between sets sc and sc' , which helps correct the mismatches. Finally, both TX and RX utilize a strong extractor to extract the same key from the agreed sc . Because sc and sc' are regarded as binary vectors, BCH codes are particularly suitable in this case and were also suggested in [1].

Chapter 3

Methodology

This chapter elaborates the theoretical framing and methodological decisions that underpin our research.

In this thesis, we present two studies that develop two primary categories of IMD pairing techniques: Study II focuses on pairing based on the user’s bodily motions (detailed in Chapter 5), and Study III explores pairing using vibration signals¹ (detailed in Chapter 6). The techniques we propose adhere to the same cryptographic framework, employ identical user study protocols (user studies were conducted to validate our proposed techniques), and utilize the same prototype design and implementations, as well as evaluation metrics. We introduce these unified methodologies in this chapter.

In Section 3.1, we introduce the complete cryptographic protocol we propose for pairing between an IMD and an external device, detailing the rationale behind the design of the framework and providing a comprehensive security analysis. In Section 3.2, we explain our methodology regarding the user study protocols, including the design and development of prototypes and the recruitment of human participants. In Section 3.3, we detail the metrics we use to evaluate the results of our user study and assess the performance of our proposed pairing techniques.

This chapter contains contents from our Article I, II and III. Note that this chapter focuses predominantly on methodological elements that are relevant across the entire thesis. For detailed information pertaining to each individual study, we refer the readers to the respective subsequent chapters.

In particular, the content in Section 3.1.2 is mainly from Dr. Mark Ryan, who is a co-author of Article I.

¹Our pairing technique using vibration is fundamentally different from previous work like [38], and offers unique advantages in the IMD context.

3.1 Framework of Our IMD Pairing Techniques

As a reminder, we aim to develop new IMD pairing methods that overcome the shortcomings identified in both academic literature (as summarized in Section 2.2) and real-world IMD implementations (presented in Chapter 4). Specifically, existing approaches are found to be (i) not perceptible enough, which makes them susceptible to injection attacks by nearby adversaries who attempt to pair with the IMD without the patient noticing, (ii) prone to eavesdropping attacks, particularly offline brute-force attacks, and (iii) challenged by deployability issues, such as prolonged pairing duration, and incompatibility with existing IMD products or off-the-shelf consumer devices like smartphones.

In this thesis, we propose a novel protocol to pair (*i.e.*, to exchange a cryptographic session key) with an IMD, which largely mitigates both injection and eavesdropping attacks, is compatible with state-of-the-art IMDs and prevalent consumer devices, and completes within several seconds.

3.1.1 The Cryptographic Protocol

The complete protocol is demonstrated in Figure 3.1. We will provide a detailed explanation of the protocol in this section.

The external device first initializes the pairing—this can be any existing wake-up process adopted by current IMD products [9]. Subsequently, two steps are executed:

- ① **OOB key exchange:** The IMD and the external device first exchange an *ephemeral and low-entropy* secret, namely Short-Term Key (STK), via an OOB channel that carries a high level of patient perception, thereby mitigating injection attacks.

This process involves both devices simultaneously taking measurements from an inertial sensor (M and M' in Figure 3.1), which has been prevalent in modern IMDs and consumer devices. Due to potential inherent noise in the sensor measurements, a fuzzy cryptographic primitive [19], such as a fuzzy extractor [1], is utilized to rectify such noise (that is within a certain threshold) and exchange the STK without revealing the secret to the adversary. In Figure 3.1, we employ the syndrome construction of the fuzzy extractor due to its low computation and energy load on embedded systems, as we will demonstrate in Chapter 7. Full details of this construction are given in Section 2.4.

In Study II and III, we propose two categories of *user-driven* OOB channels that aim to provide high patient perception to increase security. The first category of OOB

channel requires the user to hold the external device and perform several *bodily motions* (body rotation or tapping) on the patient’s body. The second category utilizes a specific physical signal—*vibration*, which naturally offers high user perception—for pairing between two devices. In Chapter 5 and Chapter 6, we will detail the design of these OOB channels, and describe our evaluation results with humans.

Both OOB channels require the user to perform a series of motions. Throughout this process, the two devices measure sensor readings, which are then utilized as a shared secret for pairing. For instance, in our rotation-based pairing method, the IMD (or the external device) records an integer *rotation angle* in degrees every time the user executes a body rotation, resulting in a sequence M (or M' for external device) $= \{m_1, m_2, \dots, m_x\}$ (where x is the total number of required motions) upon completion of all motions. Subsequently, sc (or sc') is formed by concatenating the bit representations of the data points in the motion sequence M (or M'), which is then used as input for a fuzzy extractor to exchange the STK.

- ② **PAKE:** After the IMD and the external device exchange the STK, subsequently and instantly, they execute a Password-authenticated Key Agreement (PAKE) method [78] to further exchange a *robust and high-entropy* Long-Term Key (LTK) as the session key. A concrete example of PAKE is DH-EKE. Here, the STK plays the role of the “password” in the PAKE. The prerequisite for successful PAKE execution is that both parties share the exact same STK; otherwise, the legitimate devices can detect a failure, thus abort the current session and require a restart (*i.e.*, a new STK exchange). This means that sc and sc' (see Figure 3.1) must be sufficiently similar.

The involvement of a PAKE has two major advantages. First, the protocol execution time can be reduced. Instead of exchanging the entire session key (most likely a 128-bit key), the OOB channel is used solely to bootstrap a short STK. The PAKE employs the normal wireless channel and is thus usually much faster compared to the OOB exchange.

Second, the PAKE can also significantly reduce the potential for eavesdropping: (i) It rules out offline brute-force attacks (*i.e.*, record the traffic and perform offline exhaustive search) on the OOB channel. The STK is short-lived and only valid until the start of the PAKE, which means that the adversary must obtain the STK online in a usually short timeframe. (ii) The adversary cannot endanger the patient by passive observation alone. For example, assume DH-EKE (see Figure 3.1) is used as the PAKE, then the underlying Diffie-Hellman scheme naturally prevents the adversary from obtaining the LTK by eavesdropping only. (iii) It restricts the time of attempts of a MITM adversary. Through proper implementation of a PAKE, the adversary is confined to a single attempt to run the PAKE with a legitimate

device [3]. The adversary's failure (e.g., due to possessing a wrong STK) will lead to the abortion of the current session.

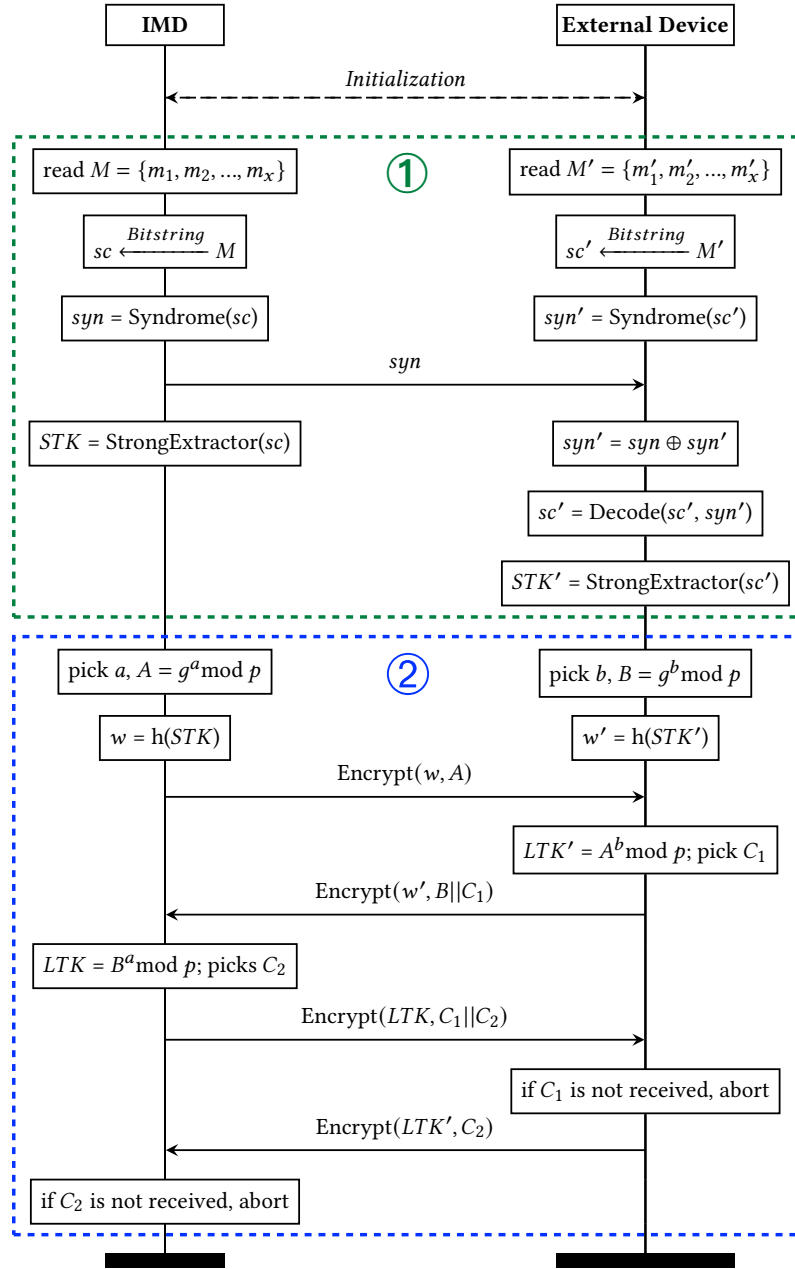


FIGURE 3.1: Our pairing protocol. ① is OOB key exchange, which involves measuring sensor data and applying a fuzzy extractor scheme (the syndrome construction, see Section 2.4 for details) [1] to remove noise from these measurements. *Bitstring* is the process of converting a set of integers into a bit string. “Syndrome” stands for the syndrome generation phase of an error correction code, while “StrongExtractor” refers to the application of a strong extractor (such as a secure hash function [2]) on the data. $\oplus$ is XOR operation. “Decode” denotes the decoding part of an error correction code. ② is PAKE (here we use DH-EKE [3] as a concrete example). g and p are public parameters. $h()$ is a secure hash function, and $\text{Encrypt}(key, msg)$ refers to the encryption of the message msg with key . Note that although we use fuzzy extractor and DH-EKE, many other fuzzy primitives or PAKE methods may also be applicable.

It is noteworthy that the required amount of entropy for STK is critical, yet it is more subjective than a unified standard in previous research [63, 68]. For our pairing protocol, considering that the adversary has only one single attempt for MITM attacks due to the use of PAKE, we believe that it is sufficient as long as the entropy of STK exceeds existing industry standards, such as 4-digit PIN codes (used in, for example, ATM machines, Bluetooth technology, and other security systems [79]) with an entropy of 13.3 bits. This would limit the adversary's success probability in brute-force attacks to less than 0.01% (assuming the adversary is limited to guessing only).

3.1.2 Security Analysis with PAKE

We believe that our proposed OOB channels can effectively mitigate injection attacks. The pairing can only be conducted by a legitimate patient or doctor, and a malicious activity (such as an adversary tapping on the patient's body for a few times) would be readily perceptible to the patient. Our user studies further substantiate our claim, as we will elaborate in Chapter 5 and Chapter 6.

Therefore, in our protocol, the adversary is forced to wait for the patient or doctor to perform the STK exchange. Concurrently, they need to eavesdrop/guess (but not inject) the STK and execute the PAKE with one legitimate device. Here, we quantify the success probability of the adversary, denoted as P_{adv} . For simplicity, we assume that the adversary aims to impersonate the external device and connect with the IMD (which is normally the worst case), but note that the situation is similar if the target is the external device.

The pairing protocol might fail under some circumstances, thus, should tolerate a certain number of rounds (*i.e.*, restarts). The maximum tolerable number of rounds can be fixed by design, denoted by n . In each round, let P_c denote the adversary's success probability to eavesdrop/guess the STK. We assume that if the adversary can obtain the STK, they can use it to complete the PAKE, and thus, the adversary's success probability is also P_c . Let Q_c be the probability that the legitimate external device succeeds in completing the PAKE with its exchanged STK. Given this, since the successes of the external device and the adversary are *exclusive* but not independent, the probability that neither of them obtains STK (and hence the key exchange fails) is $1 - P_c - Q_c$. There are three possible outcomes per round:

- The external device succeeds.
- The adversary succeeds.
- Both fail and the protocol proceeds to the next round.

Note that success of both parties are impossible because the IMD can do PAKE only with one device. We estimate P_{adv} in the following.

Proposition 3.1. *Given an adversary that conducts attack for at most n times. For example, the adversary succeeds if they succeed in the first round; or both fail in the first round, and the adversary succeeds in the second round; etc.*

$$P_{adv} \leq P_c + \phi P_c + \cdots + \phi^{n-1} P_c = \frac{P_c(1-\phi^n)}{1-\phi}, \text{ where } \phi = 1 - P_c - Q_c.$$

Proof. The first term corresponds to the case that the adversary succeeds in the first round; the second term corresponds to the case that the adversary succeeds in the second round; and so on. Note that these cases are exclusive, and therefore can be summed. The adversary succeeds in the n th round if they fail in the first $n - 1$ rounds (with probability ϕ^{n-1}) and succeed in the round after that (with probability P_c). $\square$

Furthermore, we propose that P_{adv} is always less than some constant multiple of P_c .

Proposition 3.2. *There is a constant k representing the legitimate external device's ability that is independent of P_c , such that:*

$$P_{adv} \leq k \cdot P_c, \text{ when } k = \frac{1}{Q_c}.$$

Proof. Since $1 - (1 - P_c - Q_c)^n \leq 1$ (remember that these are probabilities, hence in the interval $[0, 1]$), we have $P_c(1 - (1 - P_c - Q_c)^n) \leq P_c$; also $P_c + Q_c \geq Q_c$, and therefore

$$P_{adv} \leq \frac{P_c(1 - (1 - P_c - Q_c)^n)}{P_c + Q_c} \leq \frac{P_c}{Q_c} = k \cdot P_c$$

when we put $k = \frac{1}{Q_c}$. $\square$

To understand this proposition, let us assume, e.g., the legitimate external device usually succeeds ($Q_c > 0.9$), and the adversary usually fails ($P_c < 0.05$), although the adversary has a substantial probability of causing the protocol to fail and have to be restarted. In this case, we see that $P_{adv} < 0.056$. The situation improves if a more qualified OOB key exchange is adopted, e.g., the legitimate device is more likely to succeed ($Q_c > 0.9$) and the adversary is more likely to fail ($P_c < 0.05$), then we have $P_{adv} < 0.056$. This shows that the advantage the adversary gets of being able to force a restart of the protocol is fundamentally limited.

Note that if the adversary interferes with the legitimate external device (*i.e.*, to reduce Q_c) and causes repeated protocol restarts (essentially a Denial-of-Service attack), the external device can alert the user, who can then take actions to investigate or report the suspected attacks. This can further decrease P_{adv} .

3.1.3 Potential Attack Vectors

Here, we specify the potential types of attacks that are particularly relevant to our proposed pairing protocol. It is important to note that the attack vectors here is based on, and does not contradict, but rather supplements, the adversary model we established in Section 2.1.1.

We first consider the possible adversary goals: **(G1)** The adversary wants to impersonate the legitimate external device and pair with the IMD to access the patient’s medical data and deliver malicious therapies. **(G2)** The adversary wants to impersonate the IMD and pair with the legitimate external device to e.g., cause the IMD to not receive important commands from the doctor. **(G3)** The adversary wants to recover the shared secret between legitimate devices and then uses it to eavesdrop and/or manipulate the communication between legitimate devices. These lead to the following attack methods relevant in the particular context of our pairing techniques.

Impersonation attack: The adversary uses a data sequence M'' (recall that M and M' represent the data sequences held by the IMD and the external device prior to executing a fuzzy primitive, see Figure 3.1), in an attempt to impersonate a legitimate device. They could achieve (G1) or (G2) if their M'' closely matches IMD’s M or external device’s M' , respectively.

We consider two threat scenarios where the adversary has increasing capabilities. *Not knowledgeable of victim:* The adversary does not possess specific knowledge of the user’s habits, but can use a M'' previously collected from other users or even themselves. *Knowledgeable of victim:* The adversary possesses the user’s historical information, in particular the user’s past M or M' , to perform attacks.

Brute-Force attack: The adversary brute-forces the STK and launches MITM attacks to achieve (G3). As mentioned, the use of PAKE eliminates offline brute-force attacks, *i.e.*, the adversary eavesdrops the pairing traffic and performs offline analysis in order to crack the secret after pairing.

For our protocol, the adversary can only try an *online attack*: During the pairing process, the adversary tries every possible STK until they hit a correct one. This attack is limited by the randomness level of STK (such as its entropy), and the rate at which an adversary can run multiple pairing sessions with the target device (only one attempt in our case).

Side-Channel Attack (beyond the scope of this thesis): The pairing techniques we propose leverage users’ physical behavior as secrets, and thus are naturally and inevitably vulnerable to advanced side-channel eavesdropping using cameras and computer-vision techniques, or microphones for motion-generated noise. This vulnerability is

shared by previous pairing techniques that involve physical operations (including manual password input in, e.g., ATM machines or PCs) [62, 63, 65, 68, 69], and OOB channels for IMDs (e.g., those using audio [22], vibration [38, 41], and physiological signals [32, 33, 46]).

We will discuss the impact of side-channel attacks and their mitigation strategies in Section 8.4.

3.2 User Study

In this thesis, we propose user-driven pairing methods for IMDs, and the completion of the pairing process requires the user to perform certain actions. Therefore, developing prototypes of our proposed techniques and evaluating them with human participants are essential to assess their practical applicability.

In this section, we will elaborate our methodology regarding the user study protocols, which are followed by our Study II and III that involve human participants. In particular, we will describe the implementation of the prototypes and the recruitment of participants. It is noteworthy that the information provided in this section may not be exhaustive, and additional details (such as development of certain algorithms) may be included in subsequent chapters when discussing specific experiments related to individual proposed pairing techniques.

3.2.1 Prototype Implementation

IMDs are implanted inside patients' bodies. The ideal way to test the feasibility of an IMD pairing approach would be to recruit real IMD patients, program their IMDs, and evaluate the pairing performance. However, due to the potential life-critical risks to the patients, such experiments are not ethical and are also not adopted in previous research. Therefore, it is crucial to develop appropriate prototypes that accurately simulate the IMD scenario without intruding on or affecting the participants' health. Not surprisingly, this simulation is often a significant challenge in the research community for IMD security.

In the following, we introduce the prototypes we have developed for our studies. Although there may be compromises in certain areas, such as using a meat bag to simulate the human body, we believe that our prototypes can effectively replicate the scenario of IMD patients, and provide valuable references for future IMD-related research.

The IMD. There is a wide variety of IMDs available worldwide. For validation purposes in a laboratory setting, it is neither feasible nor necessary to develop simulations for each single type. In our work, we assume that the IMD is a *pacemaker* implanted beneath the chest, which is one of the most commonly used IMDs in practice [80]. However, it is important to note that the pairing techniques proposed in this thesis are not dependent on the specific type of IMD or its location of implantation. For example, we believe that these techniques would also be applicable to an insulin pump implanted beneath the waist area.

In our prototypes, we use one inertial sensor to represent a pacemaker. For Study II, we employ an Adafruit Bosch BNO055 inertial measurement unit (utilizing its integrated accelerometer and gyroscope) [81]. For Study III, we use an InvenSense triaxial MPU-6050 accelerometer [82]. Both sensors are commonly used in real-world embedded systems and have been utilized in prior research [83].

The IMD prototypes we use during the user study are shown in Figure 3.2a and Figure 3.3a. In both studies, the sensor is housed inside a 3D-printed case that simulates the shape of a pacemaker (the pink component in Figure 3.2a and the yellow component in Figure 3.3a). An Arduino Nano 33 BLE board is connected to the sensor via a cable, allowing it to interface with the sensor and record data on an SD card for analysis. This board features a 32-bit Cortex-M microcontroller and closely resembles the capabilities of an IMD [8].

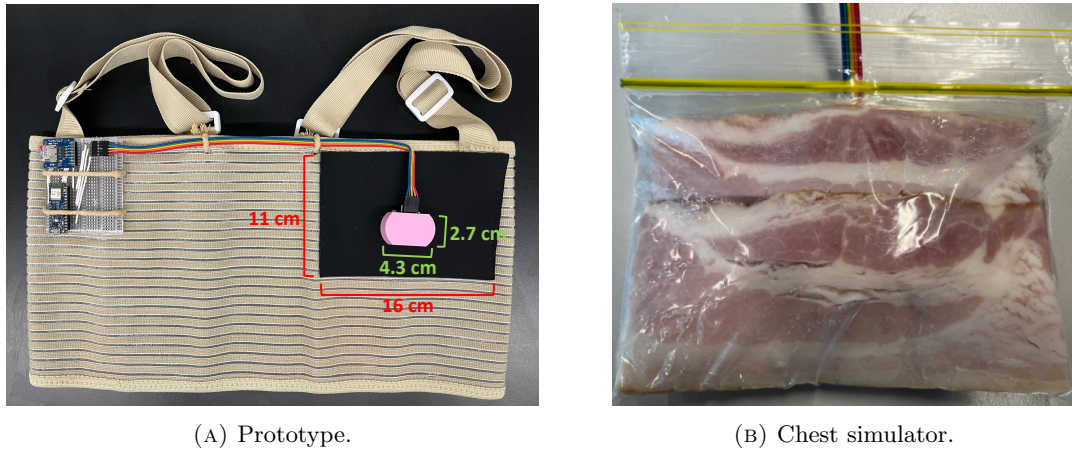


FIGURE 3.2: Experimental setup for Study II (Chapter 5).

The external device. We assume that the external device is a *smartphone*. As mentioned, nowadays off-the-shelf consumer devices are of increasing importance to give patients (some) access to their IMDs. For instance, Medtronic, a global leader in medical devices, has incorporated BLE into IMDs [9, 10], and developed a smartphone app that allows the patient to view some basic characteristics of their IMDs over a BLE link [11].

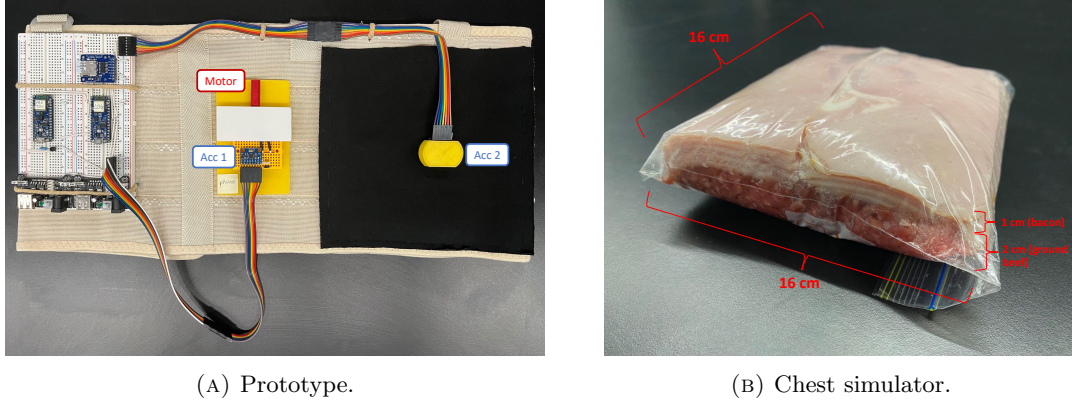


FIGURE 3.3: Experimental setup for Study III (Chapter 6).

Note that smartphones cater to general use cases; thus, if the pairing method works with a smartphone, it should also be compatible with proprietary devices developed by medical organizations.

For Study II, an inertial sensor is needed at the external device side, which is commonly available in modern smartphones. We use a Google Pixel 6 smartphone as the external device for our study and develop Android apps that guide users through the pairing process. The smartphone connects wirelessly to the IMD via a BLE link.

For Study III, we require the external device to have a vibration motor capable of emitting signals at varying frequencies. Therefore, we do not directly use a smartphone as the external device, because the most common operating systems on mobile devices—Android and iOS systems—do not provide an API interface for direct control of the vibration motor frequency.

Instead, we use an Eccentric Rotating Mass (ERM) type vibration motor, which is commonly found in commercial portable devices like smartphones, along with another MPU-6050 sensor (Acc1 in Figure 3.3a) to simulate an external device. These components are mounted on a plastic cuboid board measuring $11\text{ cm} \times 7\text{ cm} \times 0.5\text{ cm}$, which mimics the size and shape of a typical smartphone. The mock-up also includes a white handle to assist participants in holding the device. A separate Arduino Nano 33 BLE board controls the vibration motor and the accelerometer. Specifically, the Arduino board can adjust the vibration frequency by supplying different driving voltages using the Pulse Width Modulation (PWM) technique.

The body (chest) simulator. Given that pacemakers are embedded inside the body, it is important for our experiments to mimic an environment that resembles the human chest. Drawing inspiration from the experiments in [16, 22, 38], we use a 1 cm layer of bacon and 2 cm layer of lean ground beef to replicate the chest’s physical properties (see Figure 3.2b and Figure 3.3b). The 1 cm depth is a standard depth for pacemaker

implantation [84]. In our study, we embed our pacemaker simulator within the meat layers, which are kept inside a food storage bag at room temperature. This bag of meat is subsequently placed in a pocket stitched onto an elastic chest band, positioned around an area corresponding to the human heart’s location (the black pocket in Figure 3.2a and Figure 3.3a). Participants were asked to wear the chest strap throughout the user study to mimic the conditions of pacemaker users. The meat was replaced at the beginning of each day to prevent spoiling.

3.2.2 Participant Recruitment

Our user studies received approval from the ethics committees of the institutions associated with the research, including the University of Melbourne, Australia, and University of Birmingham, UK.

For both Study II and Study III, we first conducted a pilot study with six individuals (aged 22 to 32, comprising four females and two males) to study their attitudes towards our proposed pairing methods and to identify and rectify any issues with our experimental setup. In the main study for both, we recruited 24 participants, including 11 males and 13 females, with ages ranging from 21 to 52.

Specifically, for Study II, given that pacemaker users are often seniors [48], we also ran a co-design workshop with two senior individuals who had intimate knowledge and experience with pacemakers: (i) a 74-year-old female cardiology nurse with decades of experience in assisting pacemaker patients, and (ii) an 81-year-old male pacemaker patient.

The user studies were carried out in an enclosed room within the User Experience lab at the University of Melbourne. Participants were recruited through online advertisements posted on the student and staff hubs of the University of Melbourne, and each participant received AUD 30 as compensation for their time. All data collected in the user study were specifically processed to ensure anonymity and untraceability of identity. They are stored securely in the University of Melbourne’s cloud drive, and will be deleted automatically in five years of the study.

3.3 Evaluation Metrics

To comprehensively and uniformly evaluate our proposed pairing techniques in this thesis (Study II and Study III), we establish a series of metrics focusing on accuracy, security, usability, and time efficiency.

3.3.1 Accuracy

Typically, the accuracy of a pairing system is measured using *False Rejection Rate (FRR)* and *False Acceptance Rate (FAR)* [63, 68, 73]. (i) FRR is the frequency at which legitimate IMD-external device pairing is incorrectly denied. A high FRR is likely to lead to poor usability. (ii) FAR indicates the frequency the system mistakenly authorizes pairing between a legitimate IMD (resp. external device) and a malicious external device (resp. IMD). FAR gauges the resilience of pairing against impersonation attacks (see Section 3.1.3), and a high FAR could lead to poor security.

During the pairing process, there is often mismatch (denoted by d) between the readings of the IMD and the external device (M and M' in Figure 3.1) due to inherent noise in sensor measurement and user behavior (e.g., user hand wobbles when holding the external device). However, a successful pairing requires that two devices share the exact same secret. Hence, a fuzzy cryptographic primitive [19, 68] is utilized in this case to correct the mismatch without revealing the secret. At the core of this method is the selection of a threshold (denoted by Thr): the mismatch can be rectified (and thus the pairing is accepted) if $d \leq Thr$; otherwise, the pairing is rejected. A false rejection occurs when $d > Thr$ for a legitimate data pair, and false acceptance when $d \leq Thr$ for an illegitimate pair.

As such, one can balance FRR and FAR by adjusting Thr . The Equal Error Rate (EER), where $FRR = FAR$, has been used in certain applications for striking a balance between these two aspects [85]. However, because security is vital in the context of IMDs, we set a smaller Thr to ensure $FAR = 0$ for optimal security, and use the corresponding lowest FRR to represent accuracy, aligning with previous work [68].

Given $FAR = 0$, what FRR value is acceptable? This is a more subjective than unified standard in the research community. Previous works have suggested using an FRR threshold of 10% for pairing systems [65, 68], *i.e.*, the probability of a legitimate user successfully completing the pairing in one attempt is no less than 90%. Our work aims for better pairing accuracy: we consider an FRR threshold of 5% as indicative of good performance, denoting a legitimate user's success rate of higher than 95%.

3.3.2 Security

The FAR metric evaluates the system's security against impersonation attacks. On the other hand, the resilience against brute-force attacks (see Section 3.1.3) is determined by the level of randomness in the data type, which can be gauged in two primary ways: (i) Through the U.S. National Institute of Standards and Technology (NIST) statistical

test suite (NIST Special Publication 800-22) [86] that offers a comprehensive randomness assessment of a random number generator (RNG), a method widely recognized within the cybersecurity community [34, 68]. (ii) By measuring entropy, which quantifies the amount of information contained in each motion event. In this thesis, we employ the Shannon entropy, a method extensively used in prior work [33, 65, 83, 87]. The Shannon entropy of a random variable x is computed as $E = -\sum_{i=1}^k P(x_i) \log P(x_i)$ where k denotes the total number of possible x outcomes. In our work, the base of the $\log$ operation is 2, thus the unit of the entropy value is the bit.

For our proposed pairing techniques, we require the secret shared through user behavior to pass the NIST statistical test suite. As mentioned earlier, the entropy contained in this shared secret should at least exceed that of a 4-digit PIN code, a typical existing industry standard.

It is important to note that while we use Shannon entropy in our work to estimate the entropy level of a random number generator, it may also be useful to consider the min-entropy, which represents the entropy in the worst-case scenario. To address this, we have additionally applied the NIST Special Publication 800-90B test suite [88] to our data, which provides a thorough analysis of the min-entropy.

3.3.3 Usability

To evaluate the usability of our system, we use a standard SUS questionnaire, which has been widely accepted by the research community [89]. The questionnaire comprises ten items and uses a five-point Likert scale, ranging from ‘strongly disagree’ to ‘strongly agree’. The results of an SUS questionnaire can be converted into a numeric range from zero to 100. A score exceeding a threshold (usually 68 [89]) suggests good usability. The complete list of questions is provided in Table 3.1.

TABLE 3.1: Our SUS questionnaire.

I think that I would like to use this pairing method frequently.
I found the pairing unnecessarily complex.
I thought the pairing method was easy to use.
I think that I would need the support of a technical person to be able to pair.
I found the various functions in this pairing method were well integrated.
I thought there was too much inconsistency in this pairing method.
I would imagine that most people would learn how to pair very quickly.
I found the pairing method very cumbersome to use.
I felt very confident using the pairing method.
I needed to learn a lot of things before I could get going with this pairing method.

Furthermore, we conducted semi-structured interviews with the participants and held a co-design workshop with pacemaker experts to gain substantial insights into usability. The full interview questions for Study II are provided in Table 3.2, and for Study III in Table 3.3. The questions vary slightly between the two studies due to differences in experiment design and content.

The conversations with participants were recorded during the user study. Subsequently, they were analyzed through speech-to-text tools and thematic analysis protocols.

TABLE 3.2: Our interview questions for Study II.

In general, can you share your preference between the two pairing techniques based on rotation and tap, and why?
Can you share your preference between the two prototypes of each pairing technique, and why?
Have you noticed anything uncomfortable in the pairing processes?
(Given the context of how Medtronic implements the pairing between an external device and a pacemaker, which is a summary of Chapter 4) would you prefer the convenient, but invisible solutions currently adopted by state-of-the-art medical companies, or would you prefer the more interactive, yet demanding solutions that we propose?

TABLE 3.3: Our interview questions for Study III.

Can you share your experiences of using the pairing method?
Have you noticed anything uncomfortable in the pairing process?

3.3.4 Time Efficiency

The pairing time can significantly impact both usability and safety (e.g., availability in an emergency). In our studies, we record the time taken for participants to complete the pairing processes. We use the average bit rate (in bits per second) as the metric of time efficiency, which is calculated by dividing the entropy of the STK by the average time taken to exchange the STK, similar with previous work [68, 83].

Chapter 4

Case Study of a State-Of-The-Art Pacemaker

In this chapter, we present our Study I: analyzing a modern pacemaker—the Azure model from Medtronic [8], which is one of the most commonly used IMDs worldwide—as a case study to understand how today’s IMDs operate and to analyze potential issues. This encompasses a range of research work, including but not limited to, collecting relevant information about this pacemaker model, visiting a cardiology clinic to observe the pairing process, and conducting a security analysis.

The initial objective of this study was to reverse-engineer the pairing protocol of the Azure pacemaker model and identify potential vulnerabilities. This required access to the corresponding device programmer, which was only available at hospitals and cannot be rented or purchased. Therefore, we asked for help from a hospital in the UK, and a cardiology clinic gave us temporary access to a demo setup (without any patient involvement or contact) to conduct the study. Unfortunately, due to the impact of the COVID-19 pandemic, the situation at the clinic became worse, and we were forced to abandon future studies.

Nevertheless, we have gained valuable insights from the study we conducted. For example, our work led us to a better estimation of today’s IMDs’ capabilities, and allowed us to observe real-world vulnerable design issues in pairing with IMDs. These findings directly contributed to our further studies outlined in the following chapters.

We note that the work in this chapter focuses on the discussion of general pairing design principles, such as whether the pairing process should be perceptible or imperceptible, rather than on specific vulnerabilities that might be exploited by an attacker to

threaten patient safety, which would require responsible disclosure and further ethical considerations.

4.1 Hardware Analysis

The Azure model pacemaker (see Figure 4.1a) is developed by Medtronic (a global leading medical device company), and is one of the most commonly used IMDs worldwide [8]. We begin by analyzing its hardware capabilities. Its underlying System-on-chip (SoC) is the Dialog DA14580 [4], a widely used SoC that supports BLE communications and is utilized in many IoT devices and other IMDs, such as glucose meters developed by Vivachek [90].



FIGURE 4.1: Medtronic Azure pacemaker (A) and its programmer device (B & C).

Table 4.1 lists this pacemaker’s hardware capabilities in the most important aspects. The computations are primarily handled by a 32-bit Cortex-M0 microcontroller, equipped with 42 kB of SRAM and 84 kB of ROM. These specifications are similar to other low-power 32-bit embedded systems, such as those developed by ST Microelectronics [91]. The SoC also features an AES-128 bit encryption hardware accelerator to facilitate the encryption of wireless communications, and supports standard Bluetooth 4.2 technology. Additionally, this SoC is well-known for its low power consumption and the ease with which proprietary communication protocols can be developed based on its provided Bluetooth stack firmware.

TABLE 4.1: Hardware capability of Medtronic Azure’s SoC [4].

Hardware type	Parameter
Processor	32 bit ARM Cortex-M0
SRAM	42 kB
ROM	84 kB
Security processor	AES-128 bit encryption accelerator
Wireless standard	Bluetooth 4.2

Pacemakers are highly sensitive to energy consumption. Unfortunately, there are no published reports about the battery capabilities of the Azure pacemaker. Therefore, we

can only provide a preliminary estimation. The energy density of a pacemaker's Li/I₂ battery can reach up to 210 Wh/kg [60]. Assuming that the battery constitutes 80% of the total weight of the pacemaker, which is about 20 g for this model, the energy capacity would be approximately 4 Wh (14,400 J).

Modern pacemakers are equipped with inertial sensors for medical purposes [57, 59]. For instance, an accelerometer is used to enable Medtronic's Rate Response feature, *i.e.*, adapt the pacemaker's pacing rate to changes in the patient's physical activity [58]. However, similar to the battery capacity, we have not yet found reports detailing the exact capabilities of the inertial sensors in these devices, such as their sampling rates.

4.2 Study of the Pairing Process

We study the pairing process of Azure pacemakers through exhaustive literature searching (including grey literature) [8, 9], as well as multiple visits to a cardiology clinic and discussions with doctors. The Azure pacemaker primarily supports wireless connections with two main types of external devices.

- The *programmer* device (see Figure 4.1b): This device is securely kept in hospitals and is not available for sale or rent. It has full access to the pacemaker, including the ability to read historical medical data and to change or disable therapies.
- A *smartphone* with Medtronic's "MyCarelink" App [11]: This connection allows only the relay of medical data from the pacemaker to the hospital's server. All data are first encrypted using a key shared between the pacemaker and the smartphone, then sent to the smartphone, and finally transmitted to the registered hospital server [9].

It is noteworthy that, the secure communication channel between the pacemaker and a smartphone is not established via a traditional pairing process that involve only these two entities. Instead, the patient must first visit the clinic to allow a doctor to pair with their pacemaker using a device programmer. Subsequently, the doctor can register the patient's smartphone with the hospital's server. Only then is a secure encryption key bootstrapped between the pacemaker and the smartphone. Therefore, in this thesis, we omit this scenario and only focus on the pairing process between the pacemaker and the programmer.

4.2.1 The Pairing Protocol

We conducted an analysis of the pairing protocol at the cardiology clinic of the Queen Elizabeth Hospital Birmingham in the UK, under their permission in advance. We requested a cardiologist to complete the pairing process between their programmer and a patient’s pacemaker. During this process, we utilized an Adafruit Bluefruit LE Friend dongle [92] to capture the Bluetooth communication traffic, which was subsequently analyzed using Wireshark [93]. Our analysis led to the following findings:

First, the pacemaker is designed to optimize battery life by entering sleep mode for the majority of the time. Every three minutes, it briefly wakes up to enter Bluetooth advertising mode and check for any external devices seeking communication with it. If not, the pacemaker will return to sleep mode to conserve energy.

Second, the pairing between the pacemaker and the programmer uses a proprietary protocol that builds upon the standard BLE pairing process, specifically employing the Legacy Pairing mode with the “Out of Band” (OOB) security level [94]. In our scenario, the Azure pacemaker and the programmer share a 175 kHz wireless channel (separate from the conventional 2.4 GHz BLE channel) as an OOB channel¹ to exchange an ephemeral secret. Once this secret is successfully exchanged, the devices proceed with the BLE pairing protocol, leveraging the previously exchanged secret to further establish a robust and secure session key.

We have not evaluated the security of this 175 kHz channel against remote eavesdropping or injections. However, it is important to consider that a determined adversary with access to high-power antennas and sophisticated equipment might potentially compromise this channel. Future research and testing are necessary to assess this channel’s robustness, and to identify any potential vulnerabilities that could be exploited by attackers with advanced abilities.

4.2.2 Patient Perception

During the study in clinic, we not only captured the communication traffic to examine the underlying cryptographic protocol, but also observed the pairing process in person. This allowed us to evaluate the pairing from an end user perspective, gaining a firsthand understanding of how the pairing works.

¹Technically, the 175 kHz channel is still a radio frequency (RF) channel. However, this channel is deemed more secure than the Bluetooth channel, because it is more difficult to be eavesdropped on due to its lower frequency and range.

We found that the pairing process in practice was rather straightforward: the doctor simply moved the programmer head (about the size of a smartphone, see Figure 4.1c) within close proximity (less than ten centimeters) of the pacemaker and pressed a button. After a short period, the pairing was completed automatically, enabling the doctor to access the patient’s cardiac status history on a computer or send commands to adjust the pacemaker’s therapy.

This procedure was very efficient and convenient. However, we observed that it lacks feedback cues. This makes it difficult for the patient to recognize when the pairing is taking place unless they paid close attention, and therefore, could potentially give an adversary the ability to perform attacks silently. This vulnerability is especially evident in public spaces. For example, on a crowded bus, it may be difficult for the patient to recognize if a nearby person is trying to pair with their pacemaker. In such scenarios, proximity to the patient could potentially be overlooked by the patient. Unfortunately, to the best of our knowledge, many other pacemakers, or other types of IMDs in the market also share the same issue [10, 13, 16, 30].

As a reminder, additional visual or audio cues can certainly be integrated into the external device to enhance patient perception of the pairing process. However, these cues would not be essential to the underlying cryptographic protocol (pacemakers and other types of IMDs do not have cameras or microphones to sense these signals). As such, an adversary can develop an external device that mutes all such cues, and thus may carry out an unauthorized pairing without the patient being aware.

4.3 Conclusion

In this chapter, we conducted a case study on a state-of-the-art pacemaker widely used worldwide, serving as a representative example of modern IMDs. We examined its hardware capabilities, which provided us with valuable insights for developing new and deployable pairing methods. Our analysis of the pairing protocol revealed that the pairing process is highly imperceptible, potentially exposing vulnerabilities where adversaries could silently inject secrets into the IMD, which can lead to severe consequences. The same issue exists in previous academic proposals, as we have discussed in Section 2.2.

We did not carry out an extensive experiment on detailed protocol analysis, reverse-engineering, or vulnerability discovery for this pacemaker model, primarily due to the impact of COVID-19 pandemic on our research. Nonetheless, the findings we obtained are valuable and serve as a foundation for our work, which will be further elaborated in the upcoming chapters.

Chapter 5

Using Bodily Motions to Pair with IMDs

In this chapter, we present our Study II: exploring the use of human bodily motions as a means for secure pairing with IMDs. Specifically, we propose a pairing process between an external device and an IMD that can only be completed after the user performs a series of bodily motions on the patient’s body. Note that the user in this context can be the patient themselves or a trusted third party, such as a doctor, medical staff, or caregiver.

In this chapter, the data from our user study and software used in our prototypes are currently in a temporary anonymous link: https://osf.io/cnk8j/?view_only=07a18793df66460fa1c946c8e9e0dd09. We plan to publish them properly under a permissive open-source license after we publish our Article I and II.

5.1 Motivation

This study builds on the observations and motivations outlined in Section 2.2 (for academic proposals) and Section 4.2.2 (for real-world medical devices). We aim to address a concern in the design of many existing IMD pairing methods: the lack of *patient perception* during the pairing process. This issue transcends individual protocol or hardware flaws (such as those listed in the Common Vulnerabilities and Exposures (CVE) program [95]) and resides at a higher design level with a more profound impact, questioning whether the pairing process should be “seamless” or “seamful”.

A “seamless” pairing prioritizes simplicity and ease of use. However, as highlighted before, in public and crowded environments like a bus, it may be difficult for the patient

to recognize if a nearby person is trying to pair with their IMDs. In such scenarios, proximity to the patient (required by current IMDs in the market), or even brief physical contact with them (a necessary step in many previous pacemaker pairing techniques from academia [28, 29, 33, 34, 38]), could potentially be overlooked by the patient, leading to severe consequences.

In this thesis, we advocate for a shift towards a more “seamful” design approach in IMD pairing, prioritizing security through enhanced patient perception. In this chapter, we propose a solution that includes specific bodily motions on the patient’s body as a necessary part of the pairing process. This approach is grounded in the intuitive notion that if the pairing process requires physical interactions with the patient, it becomes highly noticeable to them. Consequently, the patient is empowered to detect and halt any unauthorized pairing attempts by simply stopping the motions and seeking assistance.

It is important to note that in scenarios where the patient is unconscious, such as during emergencies, the protocol should allow for anyone with physical access to the patient to pair with their IMD. This is based on the premise that in such critical situations, there are more direct ways for adversaries to harm the patient than through IMD hacking. In these instances, the patient’s explicit consent is not required, as their immediate safety and well-being are the paramount concerns, mirroring the approach taken in other emergency medical treatments.

As we mentioned in Section 2.3, many previous studies have demonstrated the potential of utilizing human motions for pairing purposes with IoT devices [62, 63, 65, 68, 69]. However, there has been a lack of research specifically addressing the use of user motions for IMD pairing. This domain presents several unique challenges compared to prior work:

- First, the pairing motions should be simple to learn and carry out for patients, who may have difficulty to perform, e.g., strenuous movements.
- Second, the pairing process does not require direct physical contact with the IMD.
- Third, inertial sensors, irrespective of being inside or outside the body, should be able to reliably measure the motion.
- Fourth, if the patient is unable to move (e.g., due to paralysis or in an emergency), medical personnel should still have the capability to execute the motion on the patient’s body on their behalf.
- Fifth, the selected motion should inherently exhibit a wide variety to ensure high entropy, aiming to provide security comparable to existing industry standards, such

as 4-digit PIN codes with an entropy of 13.3 bits. The limited selection of motion types suitable for patients further constrains the entropy level.

These challenges highlight that prior work utilizing specific bodily motions for IoT device pairing is not suitable for our context. For example, the patient may be unable to walk for dozens of steps, and the medical staff cannot help them to complete the walk in an emergency. Therefore, there is a demand for novel pairing techniques that address these issues and cater to the specific needs of modern IMDs.

5.2 Method

This section presents the design and validation of our pairing motions, the development of our prototypes (particularly Android apps running on the external device), the algorithms implemented, and the data collection process during our user study and co-design workshop.

Other methodologies, such as the general prototype setup (e.g., the sensor, chest strap and human body simulator), as well as the participant demographic and recruitment process, are detailed in Chapter 3, as they are also shared with another pairing technique proposed in the next chapter.

We inherit our assumptions mentioned in Section 3.2.1: We assume that the IMD is a pacemaker and the external device is a smartphone, and will refer to them as “pacemaker” and “smartphone” in the remainder of this chapter. We assume the pacemaker is equipped with an inertial sensor, *i.e.*, accelerometer and gyroscope.

5.2.1 Pairing Techniques

We first design the types of motions that are suitable for pairing with pacemakers. As a reminder, the bodily motions serve as a shared random secret only known to the pacemaker and the smartphone, which later on is used together with well-established cryptographic protocols to protect subsequent wireless communications [78, 96].

While routine physical activities such as standing, walking and standing up, etc. fit some of our requirements [97], the range of such activities is limited. For instance, assuming the motion is restricted to “stand up” and “sit down”, then the entropy for each motion event is only one bit. If two devices aim to exchange a secret as robust as a 4-digit PIN, the patient would need to repeat the motion 14 times. An alternative approach would

be the use of diverse hand gestures, but we realize that it is challenging for an inertial sensor located at the chest area to discern the nuances between different gestures.

Ultimately, we narrow down to two promising options: (i) Body rotations, inspired from the electronic compass apps prevalent in modern smartphones. (ii) Taps, *i.e.*, tap the phone against the chest. Both motions are relatively easy to perform. Intuitively, an inertial sensor should be able to measure these motions whether it is inside the patient's body or held externally. For instance, taps should create peaks in sensor readings that pick up the vibration. Moreover, if needed (such as in an emergency), medical staff can rotate the patient's body using swivel chairs or stretchers, or tap directly on the patient's chest. Furthermore, these motions, given the wide range of angles for rotation and rhythms for tap, inherently bring about a relatively high level of entropy.

We conducted a pilot study with six individuals (see Section 3.2.2) to test and refine the pairing motions, aligning with previous work [62, 63]. Participants were asked to practice each type of motion for ten minutes using our developed smartphone apps (baseline prototypes in Figure 5.1). Specifically, they were required to perform each motion in two ways varying in the level of autonomy: either by following a particular instruction, such as rotating the body to a pre-determined orientation, or by performing the motion spontaneously, like rotating to an arbitrary direction. We observed the participants during the pilot study and gathered their feedback at the end.

Overall, both motion types received general acceptance. In addition, we noticed that participants could easily and accurately rotate to a specified orientation when instructed. However, when they were asked to rotate randomly, it often led to confusion. On the other hand, spontaneous tapping was natural for people, but instructing them to tap along to unfamiliar rhythms often caused tensions. Moreover, we gained further understanding of user preferences, *e.g.*, participants commonly mentioned that rotation angles under 30 degrees were difficult to execute, and angles over 180 degrees could cause dizziness. Eventually, the two pairing techniques we propose are:

- **Rotate the full body guided by instructions from a smartphone.** In this technique, users are guided by a smartphone app to execute a series of left or right full-body rotations. Each rotation angle corresponds to a portion of the secret generated by the smartphone.
- **Tap the smartphone against the chest with a random rhythm.** To complete the pairing, users need to gently tap the smartphone against the patient's chest area several times, following a random pattern on their own. Here, the secret is the tap rhythm (*i.e.*, time intervals between taps) and originates from the user.

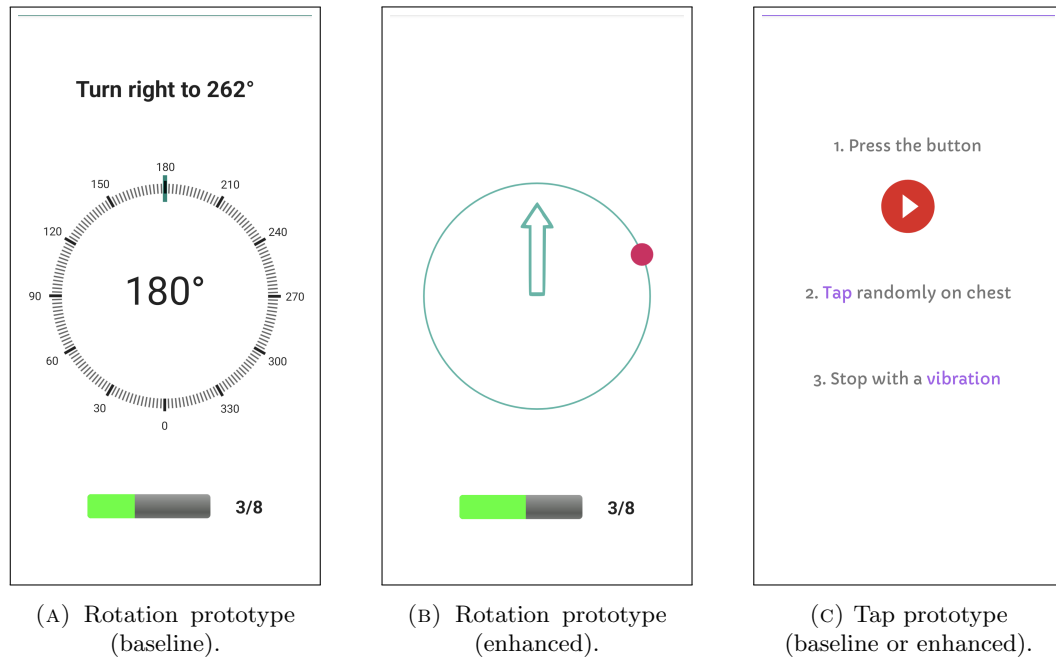


FIGURE 5.1: The smartphone user interfaces used in the study.

For each pairing technique, we develop a *baseline design* prototype that implements basic functionality mirroring existing applications on the market. Subsequently, we develop an *enhanced design* for each technique, using design heuristics aimed to improve participant performance. The enhanced versions are introduced in our experiment to ensure that our findings for each pairing technique are robust and hold across different implementations of the same pairing technique.

5.2.2 Prototypes Using Rotation

Using rotation, participants are instructed to rotate their body until they face a pre-determined orientation. Once they reach that orientation, they are given a new orientation that they need to rotate to. The orientations are picked randomly by the smartphone’s RNG.

The baseline design, as shown in Figure 5.1a, provides a classic electronic compass design commonly found in today’s smartphone application market¹. This design features a graduated 360-degree compass rose, with markings displayed at 30-degree intervals around the perimeter of the rose. A green bar, aligned with the rose, indicates the patient’s facing direction and moves smoothly along the rose as the patient rotates their body. The patient’s current orientation, in degrees, is dynamically displayed at the center. Instructions are displayed at the top of the screen, guiding the user to rotate either to the left or the right towards a specified orientation.

¹We refer to the graphic layouts of <https://github.com/Kr0oked/Compass>.

In the enhanced design (Figure 5.1b), we remove visual elements that we believe are likely to distract participants, and incorporate specific feedback designed to assist in completing the rotations. The detailed scales and markings are removed, and the compass rose is replaced by a circle. A green arrow indicates the patient's facing direction, and a red ball located on the circle represents the target orientation. As the patient rotates their body, this ball moves correspondingly around the circle. Using this prototype, the patient's objective is to ensure the green arrow points at the red ball. Upon successful alignment, the smartphone provides tactile feedback through a crispy vibration, and the red ball changes its color to green to indicate success.

Both prototypes follow several common design decisions: The target orientation is initialized in one-degree precision. Informed by observations from our pilot study, the rotation angle is restricted to between 30 and 180 degrees, in either a clockwise or anti-clockwise direction. Additionally, each rotation is instructed to be completed within a fixed 10-second time window regardless of its angle (to prevent adversaries from inferring the angle based on completion time, e.g., smaller angles might be finished faster). The 10-second duration is an intuitive choice to ensure that sufficient time is provided.

To guide users, both prototypes feature a progress bar at the bottom of the screen, visualizing the remaining time to complete the current rotation instruction. Patients must reach and maintain the target orientation by the time the progress bar is depleted. If they fail to do so, the instruction is repeated to prevent non-compliance (otherwise, the randomness of shared secret, *i.e.*, rotation angles, cannot be guaranteed). Furthermore, recognizing the potential for human error, such as hand wobbles, the app tolerates slight deviations by allowing for plus or minus two degrees from the target orientation. Finally, to inform users of their overall pairing progress, a numerical display is incorporated at the bottom, indicating the number of rotations the user needs to perform towards completion.

5.2.3 Prototypes Using Tapping

Participants are instructed to randomly and gently tap the phone on their chest using the tap-based prototypes. They are required to complete a pre-determined number of taps, at any rate they believe would maximize their randomness.

Both prototypes here are visually identical (see Figure 5.1c). The interface shows instructions that describe the pairing steps and prompt patients to tap at random rhythms. A start button allows participants to indicate when they start to tap the phone. As taps are made, the smartphone acknowledges each one and instantly responds with a brief

sound as feedback. Once enough taps are registered, the smartphone emits a vibration to indicate the completion of pairing.

The difference between the baseline and enhanced designs lies in the auditory feedback (namely the earcon [98]) they use. The baseline prototype uses one single earcon (a beep) that is played when a tap is detected, and is used to indicate to the user that the tap is acknowledged. In the enhanced design, we use multiple earcons to indicate to the user how random their tapping is. The smartphone continuously assesses the randomness of the tapping rhythm. After each detected tap, our prototype emits either a “positive”, “neutral”, or “negative” earcon—a positive earcon indicates excellent performance, a negative earcon notifies the user of a lack of randomness, and a neutral earcon is given in all other cases.

Figure 5.2 shows the workflow of the enhanced prototype. In the following, we provide full details on how we design these earcons, and how we determine which one to give after each tap.

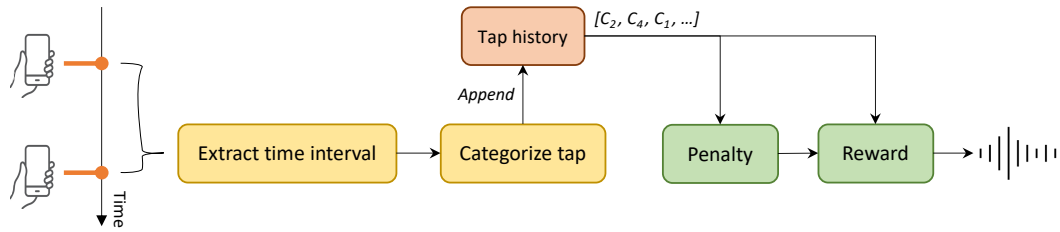


FIGURE 5.2: Workflow of enhanced prototype for taps.

Tap Categorization. First, the smartphone calculates the time interval of each tap in milliseconds, which is then categorized into discrete classes ranging from quick to slow taps. This is because human perception of time is not precise to the millisecond level, *i.e.*, it is generally easier to identify between broad time spans as opposed to closely spaced durations (e.g., 500 ms versus 501 ms). In our pilot study, participants’ taps ranged between over a hundred milliseconds to 1.8 seconds. In our work, we divide the zero to 1.8-second interval into three uniform classes and introduce an additional class for taps exceeding 1.8 seconds: C_1 : [0 ms, 600 ms), C_2 : [600 ms, 1200 ms), C_3 : [1200 ms, 1800 ms), C_4 : [1800 ms, $+\infty$). Following this, each tap is chronologically recorded in a history log.

Penalty Mechanism. At the core of this system is a penalty mechanism that alerts the user when their tapping pattern appears to be non-random. The primary challenge is to reliably evaluate the randomness of taps. For example, rhythmic tap patterns (such as drumbeats in songs) may not be considered random because of their predictability. However, such an interpretation remains inherently vague. Our work adopts the statistical test suite for randomness published by the U.S. National Institute of Standards

and Technology (NIST) [86]. This tool, widely recognized within the cybersecurity community [34, 68], is designed to determine if a RNG adheres to cryptographic standards. The test suite incorporates 15 criteria to ascertain the randomness of a bit string (*i.e.*, the output of a RNG), which we broadly group into seven categories as in Table 5.1.

TABLE 5.1: Criteria of the NIST statistical test suite. Proportion indicates how many of the total 15 criteria fall into that category.

Criterion	Description	Proportion
Oscillation	The level of fluctuation (<i>i.e.</i> , from one to zero or vice versa) among successive elements.	5/15
Frequency	The uniformity of frequency distribution between different elements.	4/15
Recurrent pattern	The frequency of occurrence of a specific block of data.	2/15
Periodicity	The uniformity in the Discrete Fourier Transform of the data.	1/15
Linear dependency	The degree to which one data block can be predicted by another through a linear relationship.	1/15
Compressibility	The degree to which data can be condensed without losing critical information.	1/15
Linear complexity	Data complexity based on its linear feedback shift register length.	1/15

The NIST test suite provides a solid foundation for our work. However, it is designed for rigorous cryptographic contexts (like encryption keys) and is inherently intricate. Its direct use to human-generated events could be overly strict. It is worth noting that our goal is not to transform patients into genuine RNGs, but to nudge them towards more random tapping than their usual spontaneous actions. Therefore, adaptations are needed to better suit our use case. In our work, we exclude the last four criteria in Table 5.1 as they are too complex. For instance, patients can be confused when they receive negative feedback, as their recent taps are “linearly dependent” on the previous ones. Moreover, these criteria also often require a large amount of data to ensure valid results, which means that many taps need to be accumulated before an analysis can be conducted. If feedback is delayed until after this collection, patients may be puzzled since they cannot remember their previous behavior. In contrast, the criteria of *oscillation*, *frequency*, and *recurrent pattern* are more intelligible. Note that they make up over 70% of the original test suite.

Another challenge is to determine the appropriate threshold that triggers a penalty for each selected criterion. For example, we regard the threshold of *oscillation* as the minimum number of consecutive taps of the same class that are considered non-random. Setting a low threshold might unduly influence user behavior, *e.g.*, penalizing after only two consecutive quick taps could lead to confusion. On the other hand, a high threshold might compromise usability because patients may not be able to recall a long history. To achieve a reasonable balance, we use Python’s secrets library [99], a widely acknowledged RNG for cryptographic applications, to simulate a million taps by randomly selecting

elements from the set $[C_1, C_2, C_3, C_4]$. The three penalty criteria (each with specific thresholds) are applied to the sequence in real-time, and a penalty is imposed if any of the criteria are met. We then run multiple iterations with different thresholds and compare the overall penalty rate, *i.e.*, the proportion of the one million elements that received penalties. Eventually, we establish the following concrete penalty strategies:

- **(P1) Repetition of identical class:** for $a \in [C_1, C_2, C_3, C_4]$, apply a penalty if the last four taps are consistently a , provided the user has executed a minimum of four taps.
- **(P2) Excessive proportion:** for $a \in [C_1, C_2, C_3, C_4]$, apply a penalty if a appears at least five times in the last six taps, given that the user has performed at least six taps.
- **(P3) Recurrence of same block:** for $a, b \in [C_1, C_2, C_3, C_4]$ and $a \neq b$, apply a penalty if the sequence of the last six taps is $[a, b, a, b, a, b]$, given that the user has conducted at least six taps.

These configurations result in a penalty rate of only 3% for Python’s secrets library. While stricter thresholds would reduce this slightly, they can have a negative impact on usability. In contrast, more lenient thresholds substantially increase the penalty rate, which should not happen on a genuine RNG. We further highlight that these strategies are independent with each other. For instance, a penalty on *P1* does not necessarily imply a penalty on *P2* and vice versa.

Reward Mechanism. Apart from penalties, rewarding patients for good performance is also crucial in guiding them towards desirable behavior. We intuitively devise a reward mechanism subsequent to the penalty mechanism. First, consistent good performance deserves a reward. Second, improved performance (increased randomness compared to prior behavior) also warrants encouragement. After rigorous design and testing within our pilot study group, we establish two concrete reward strategies. Note that a reward can be granted for a new tap that still incurs penalties—as long as it has fewer penalties than the previous round.

- **(R1) Sustained excellence:** grant a reward after six consecutive taps that have not incurred any penalties.
- **(R2) Improvement over last tap:** grant a reward if there are fewer penalties than in the previous round.

Earcon Design. We also need a scheme that converts the outcomes of both the penalty and reward mechanisms into feedback for the patient. As complex designs might induce

patients to overthink, we introduce a simplistic and straightforward scoring scheme, and limit feedback to only three types—negative, neutral, and positive, which correspond to scores of zero, one, and two, respectively. Each new tap starts with a score of one. If the patient violates any of the penalty criteria, the score is adjusted to zero. Conversely, if they satisfy a reward criterion, the score rises to two.

The last design phase focuses on choosing the most effective feedback earcons. Initially, we experimented with the sound of water drops, with varying numbers of drops representing different degrees of randomness. However, this caused significant confusion among people due to inconsistent durations of the earcon. We then explored using piano key sounds as feedback, maintaining a consistent duration but varying the pitch to suggest different randomness, e.g., a higher pitch indicating greater randomness. However, we found that users needed to familiarize themselves with the correlation between sound pitches and randomness levels, which was not ideal for usability.

Eventually, we opted to customize earcons: the positive feedback comprises two ascending notes² played on a synthesizer, which forms a rising tone that is generally associated with harmony and joy [100, 101]. The neutral feedback consists of a midrange sine wave [102] with a stable pitch³, without noticeable upward or downward movement, providing a neutral auditory backdrop. This earcon is also adopted by the baseline prototype as the (constant) feedback. Our negative feedback features a low-pitched note⁴ based on the 808 drum machine [103]. Lower frequencies have been linked to conveying sadness [101], allowing it to encapsulate negative sentiment. In our preliminary tests, participants found these earcons to be much more intuitive than the water drop or piano sounds. Furthermore, all three earcons have been crafted with a consistent and short duration of only 130 ms.

We highlight that all above designs are inherently intuitive. For instance, humans have a natural tendency towards consistent behaviors (like maintaining similar keyboard typing speeds) when using computer systems [104], which warrants reminders in scenarios demanding randomness (*P1*). On the other hand, users should be rewarded when they exhibit positive behavioral changes (*R2*). Moreover, the workflow is streamlined to minimize cognitive load on patients. Take *P3* as an example, although more complex patterns such as $[a, b, c, a, b, c]$ could occur, we exclude them for simplicity. Overall, all design details are finalized after empirical testing within the researcher and pilot groups.

²B5 to E6, 996.75 Hz to 1330.50 Hz, which forms a perfect fourth interval.

³A#5, 940.80 Hz, a pitch slightly below the first note of our positive tone.

⁴A2, 110.00 Hz

5.2.4 Algorithm Development

For rotation prototypes, tilt-compensated headings in degrees are calculated from the accelerometer and gyroscope readings using a Kalman filter [105]. For tap prototypes, we develop a real-time peak detection algorithm to identify tap events, which evaluates the norm of 3-axis accelerations ($\sqrt{a_x^2 + a_y^2 + a_z^2}$) as individual samples and screens each for potential peaks using pre-defined amplitude, peak width, and inter-peak distance thresholds. This method is lightweight and exhibited notable reliability during our study. Following this, the pacemaker calculates the time intervals between consecutive peaks, which represent the tapping rhythm, in milliseconds.

We use a Google Pixel 6 smartphone for our study and develop four Android apps corresponding to the aforementioned designs. At the start of each pairing session, the smartphone initiates an (insecure) BLE connection with the pacemaker (*i.e.*, the Arduino board). For rotation prototypes, the smartphone leverages Java’s SecureRandom API [106] to initialize each rotation instruction. The tilt-compensated heading is derived in real-time using Android OS’s “game rotation vector” virtual sensor and dynamically updated on the screen. Once the user completes a rotation (*i.e.*, reach the target orientation by the time the progress bar is filled), the smartphone logs its current orientation and signals the pacemaker to do the same. Additionally, the smartphone records the time the user initially reaches the target orientation, as a measure of the rotation’s actual duration. For tap prototypes, the smartphone synchronizes with the pacemaker at both the start and end of the pairing process. As the patient taps, the smartphone uses a peak detection algorithm (similar to that in the pacemaker simulator) to measure the time intervals between consecutive taps in milliseconds.

5.2.5 Data Collection

In the main study, participants used all prototypes, resulting in four different tasks in total. The order in which they experienced the pairing techniques (rotation or tap), and the order of prototypes within each technique (baseline or enhanced), were counterbalanced. Prior to data collection of each task, we asked participants to try our smartphone app to gain familiarity of the pairing operations. This introductory process took under a minute for all participants.

For the rotation prototypes, participants were instructed to stand and hold the smartphone while executing eight consecutive full-body rotations (see Figure 5.3a), referred to as a run. We advised them to use both hands and to avoid wrist-twisting during the rotations. Each participant carried out three runs with each prototype. The number of

rotations was deliberately set to minimize the risk of dizziness. For the tap prototypes, participants were directed to gently tap the smartphone against the black pocket region on the chest strap (see Figure 5.3b). Since tapping is less likely to cause discomfort to participants, they completed ten runs for each prototype, with each run consisting of 15 successive taps.



FIGURE 5.3: Data collection in user study.

Upon the completion of all runs on a prototype, participants were requested to fill out a standard SUS questionnaire [89] to assess the usability of the pairing method. At the end of the whole user study, we conducted an interview with each participant to gather further insights. Full details of the questionnaire and the interview are listed in Table 3.1 and Table 3.2, respectively.

Furthermore, during the co-design workshop, we asked the two senior participants to try our prototypes for only one to two runs (considering their physical conditions) and provide their opinions and advice. Their data is analyzed in Section 5.3.7.

5.3 Results

5.3.1 Participant Performance

For the rotation prototypes, recall that the rotation angles (shared secret) are randomly chosen by the smartphone. Figure 5.4a shows that the rotation angles in our experiment belong to the same random distribution as expected, *i.e.*, 30 to 180 degrees in clockwise/anticlockwise directions. The distributions show some fluctuation since the amount of data collected is limited. Figure 5.4b shows that for both prototypes, the mismatch (calculated by subtracting rotation angle reading between the pacemaker and the phone) resembles a normal distribution, with a mean close to zero and a standard deviation of 2.7 degrees. This data represents noise and error.

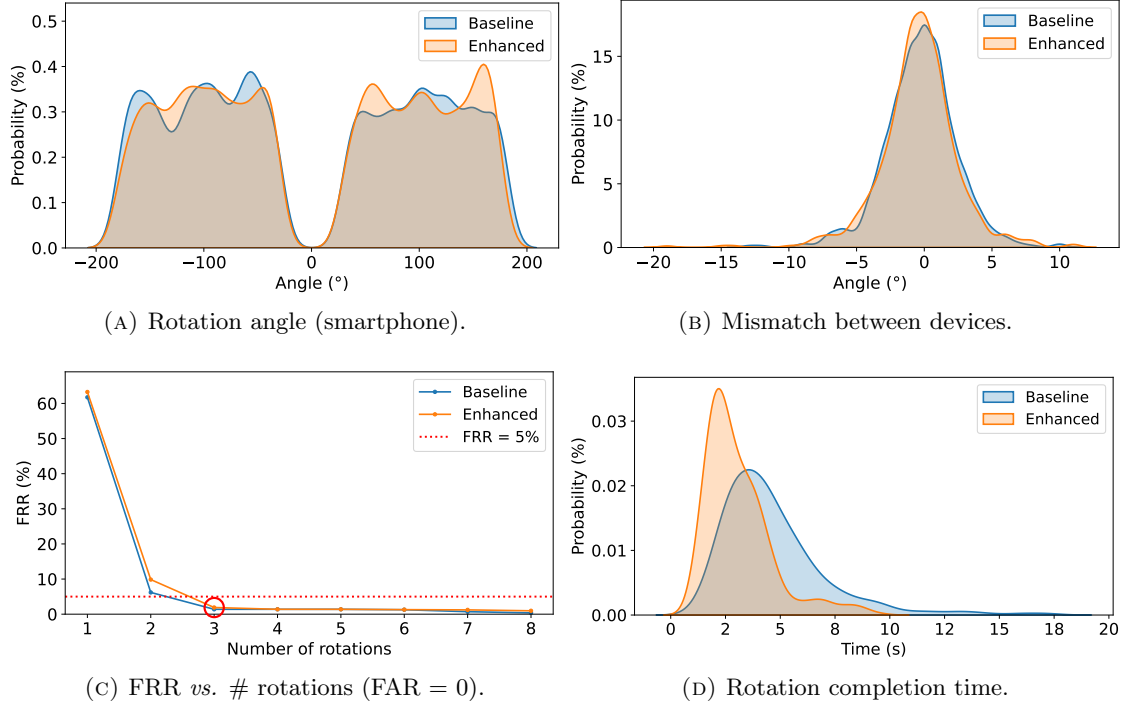


FIGURE 5.4: Results for rotation prototypes. (a) Distribution of smartphone rotation angles in baseline and enhanced prototypes. (b) Distribution of mismatch in rotation angles (*i.e.*, subtraction between a pair of angles for the pacemaker and the smartphone) in both prototypes. (c) The FRR versus the number of performed rotations, with all scenarios having zero FAR. The red circle indicates optimal configurations with FAR = 0 and FRR < 5%, using minimum required rotations. (d) Distribution of the time taken to complete each rotation in both prototypes.

For the tap prototypes, recall that the shared secret is the tap rhythm, *i.e.*, times between taps. We find that the overall time intervals for taps fall between 131 ms and 8866 ms, with little difference between the baseline and enhanced prototypes. Figure 5.5a shows the distribution from zero to 3000 ms that captures the majority of the taps. Both prototype results display distinct peaks, suggesting that users' perceptions of tapping intervals are more discrete than continuous. For instance, fast taps around 250 ms are quite common in both datasets. The data also reveals that the enhanced prototype does have an influence on user behavior, *e.g.*, for slower taps, the enhanced prototype's distribution appears more flat than that of the baseline. Our interpretation for this result is that the enhanced prototype's varied earcons encourage a more diverse range of tapping intervals, typically longer ones. Similar with rotation prototypes, the mismatch for taps appears to follow a normal distribution, with a mean nearly zero and a standard deviation of 11.1 ms.

The above indicates that both rotation and tap motions distribute across a wide range, thus possess a certain degree of randomness that is essential for pairing. Additionally, the data implies that user-induced errors and sensor noise are limited in our prototypes. It is worth noting, however, that these distributions represent aggregate behavior across

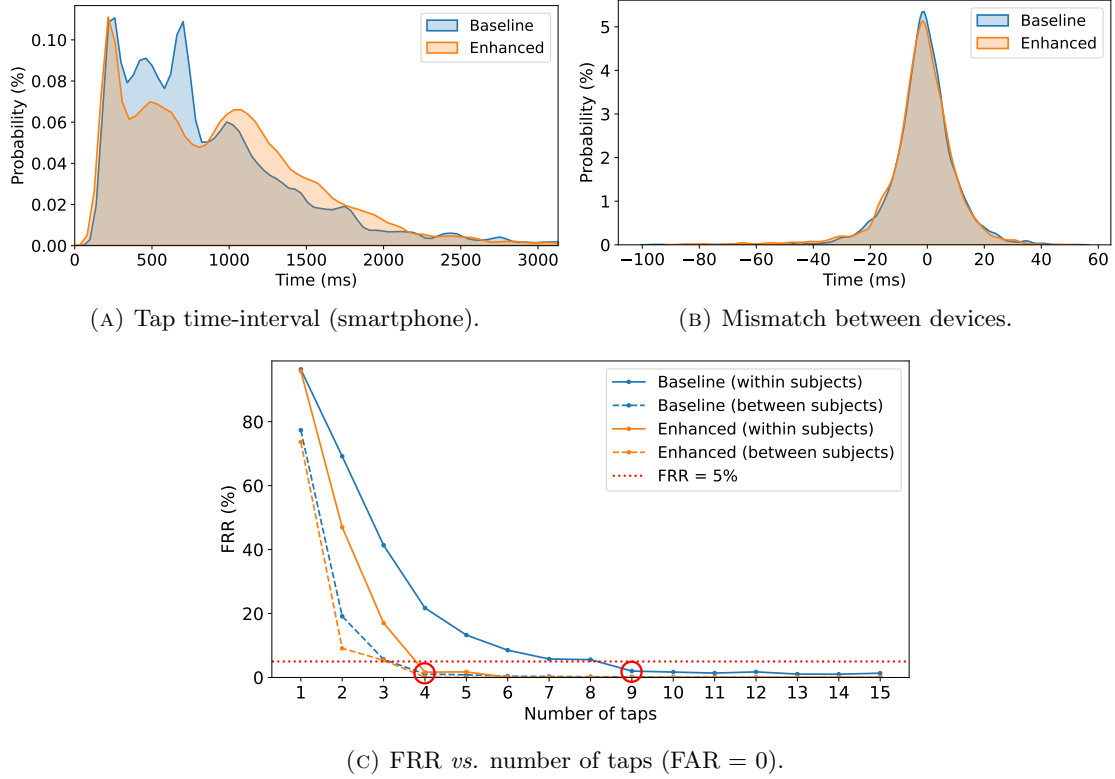


FIGURE 5.5: Results for tap prototypes. (a) Distribution of smartphone tap time-intervals in baseline and enhanced prototypes. (b) Distribution of mismatch in tap time-intervals (*i.e.*, subtraction between a pair of time intervals for the pacemaker and the smartphone) in both prototypes. (c) The FRR versus the number of performed taps, with all scenarios having zero FAR. The blue and orange lines indicate the baseline and enhanced prototypes, respectively. The solid lines represent scenarios where adversaries are knowledgeable about the users' tapping habits, while the dashed lines show cases where they are not. The red circles indicate optimal configurations with FAR = 0 and FRR < 5%, using minimum required taps.

all participants. They do not account for other important factors in the security analysis, such as the variance in consecutive motions for individual participants or differences in randomness between subjects. In the following sections, we further analyze the performance of rotation and tap pairing techniques using the evaluation metrics previously identified in Section 3.3. The full results are summarized in Table 5.2.

5.3.2 Pairing Accuracy

Pairing Based on Rotation. For each rotation prototype, we build two sets to evaluate the accuracy:

- **Set I** contains 72 ($= 24 \times 3$) legitimate pairs of rotation angles, each with a length of eight (since we collected eight angles per run). All the pairs in Set I come from legitimate pairings of the pacemaker and the smartphone.

- **Set II** comprises 72 illegitimate pairs of angles (the same amount as Set I), and is generated by randomly combining data from the pacemaker and the smartphone from different pairing sessions.

An effective pairing technique should maximize the acceptance of pairs from Set I (*i.e.*, low FRR), while minimizing the acceptance of pairs from Set II (*i.e.*, low FAR). Another important factor is the number of rotations needed—we also consider varying the length of runs, ranging from one to eight, by truncating the initial elements.

Figure 5.4c shows the performance of rotation prototypes across various numbers of rotations. As expected, an increase in the number of rotations leads to more accurate pairing. The baseline prototype displays marginally better accuracy, which we believe is because the baseline shows the target orientation in one-degree precision, while the enhanced version uses a red ball to indicate the target, covering a slightly broader range (four degrees in our setup) for user convenience. Nevertheless, the difference between the two prototypes is not substantial. Taking into account that an FRR below 5% (the red dashed line) signifies good usability (as mentioned in Section 3.3), we find that as few as 3 rotations are sufficient for accurate pairing, with $(\text{FAR}, \text{FRR}) = (0, 1.4\%)$ for the baseline prototype and $(\text{FAR}, \text{FRR}) = (0, 1.9\%)$ for the enhanced prototype.

Pairing Based on Tap. For each tapping prototype, we build three sets to evaluate accuracy. Compared to rotation-based pairing, the additional set takes into account the possible variability against non-knowledgeable and knowledgeable impersonation attacks (see Section 3.1.3)—each individual might have unique tapping patterns that could potentially be exploited, thus the access to the user’s past behaviors may offer the adversary an additional advantage⁵.

- **Set I** comprises 240 ($= 24 \times 10$) legitimate pairs, each with fifteen time intervals. All the pairs in Set I come from legitimate pairings of the pacemaker and the smartphone.
- **Set II** comprises 240 illegitimate pairs of time intervals (the same amount as Set I). It is created by randomly selecting illegitimate pairs from different participants.
- **Set III** contains 240 illegitimate pairs of time intervals (the same amount as Set I), and is generated by randomly mixing illegitimate pairs from the same participant but different sessions.

⁵In contrast, for the rotation prototypes, all motions are determined by the smartphone’s trustworthy RNG [106], making the user’s past motions independent of the current pairing session.

TABLE 5.2: Evaluation of rotation and tap pairing techniques. IA stands for impersonation attack (see Section 3.1.3). The reported entropy and data throughput values have already taken into account entropy loss due to mismatch correction between devices.

Metrics	Rotation		Tap	
	Baseline	Enhanced	Baseline	Enhanced
Min # motions against non-knowledgeable IA	3	3	4	4
Min # motions against knowledgeable IA			9	4
NIST randomness test suite outcome	not tested (limited data)		pass	pass
Entropy per motion (bits)	6.15	6.15	7.05	7.17
Time per motion (s)	4.65 (SD=2.37)	3.09 (SD=1.52)	0.88 (SD=0.65)	0.98 (SD=0.78)
Average throughput (bits per second)	1.32	2	8.01	7.32
SUS questionnaire score	62.8 (SD=21.2)	85.4 (SD=9.5)	70.7 (SD=17.3)	55.7 (SD=20.8)

In this way, *Set II* and *Set III* represent the resilience to potential adversaries who possess knowledge of the entire group of people and a specific victim, respectively.

Figure 5.5c illustrates the performance of tap prototypes with various numbers of taps. Again, more taps result in higher accuracy. When dealing with non-knowledgeable adversaries, both the baseline and enhanced groups perform well with just 4 taps, exhibiting $(\text{FAR}, \text{FRR}) = (0, 1.1\%)$ and $(0, 0.1\%)$, respectively. However, the challenge escalates when dealing with adversaries who possess deeper knowledge of a user’s tapping habits (solid versus dashed lines). In these scenarios, the baseline prototype requires a minimum of 9 taps to maintain $(\text{FAR}, \text{FRR}) = (0, 2\%)$. This result confirms the potential security risks associated with unique tapping patterns of individuals. Nevertheless, our enhanced prototype shows a significant improvement in mitigating these risks (orange versus blue lines), where only 4 taps are needed for $(\text{FAR}, \text{FRR}) = (0, 1.7\%)$. These findings suggest that our enhanced prototype successfully encourages more random tapping behavior and improves the pairing performance.

Overall, we estimate that 3 rotations are enough to defend against impersonation attacks, and the number of taps varies between 4 and 9 depending on which prototype is used. We note that for both pairing techniques, increasing the number of motions offer only marginal improvements in accuracy, but could considerably add to the user’s burden. Hence, in the following, we use these established motion numbers to assess other metrics.

5.3.3 Security Assessment

NIST 800-22 Test Suite Outcome. We do not apply the NIST 800-22 test suite on the rotation data because the randomness of rotations is guaranteed by the adopted Java SecureRandom API [106], which is widely recognized as a validated RNG for cryptographic applications.

To assess the randomness of taps generated by participants, we adopt the same approach used in previous work [34, 68, 83]: We concatenate the six least significant bits of the

time intervals from all runs performed by participants (preserving the experiment order). This bitstring (21.6k bits per prototype) is then evaluated for its randomness using the NIST statistical test suite.

The full results are given in Table 5.3. The outputs of the NIST 800-22 statistical tests are p -values that represent the probability the tapping dataset is generated by an eligible random number generator. If a p -value is smaller than a threshold (usually 0.01 [34, 68, 83]), the randomness hypothesis is rejected. Table 5.3 shows that all p -values are larger than 0.01 and hence *pass* the NIST tests. Note that we do not conduct Maurer’s Universal Statistical Test (also known as the compressibility test), because it requires a minimum data input of 387.8 kbits that is not met in our case. This test was also neglected by previous work [68, 83].

TABLE 5.3: NIST 800-22 statistical test results for taps.

Test	p -value	
	Baseline	Enhanced
Frequency	0.806	0.145
Block Frequency	0.855	0.376
Runs	0.723	0.398
Longest Runs	0.088	0.652
Binary Matrix Rank	0.706	0.065
FFT	1.000	0.628
Non-overlap Template	0.716	0.022
Overlapping Template	0.620	0.525
Universal	-	-
Linear Complexity	0.725	0.587
Serial	0.424	0.321
	0.442	0.482
Approximate Entropy	0.130	0.427
Cumulative Sums	0.854	0.285
	0.629	0.209
Random Excursions	0.738	0.419
Random Excursions Var.	0.582	0.681

Entropy Analysis. A single rotation is uniformly initialized over 300 degrees (in one degree precision) and thus contains an entropy of 8.2 bits. In contrast, a single tap in our study has an entropy of 9.4 and 9.6 bits for the baseline and enhanced prototype, respectively.

However, some entropy is sacrificed due to fuzzy cryptographic primitives that rectify mismatches between the two devices. Here, we make a preliminary estimation of the entropy loss: Using the encoding method in [68] on our dataset, the maximum bit mismatch rates (*i.e.*, percentage of different bits between two devices) for the rotation and tap prototypes are between 1.9% and 4.3%. This can be addressed by *e.g.*, a fuzzy extractor scheme with (20, 15) BCH code with a 5% error tolerance [19, 107], potentially

leading to an entropy loss of 25%⁶. In this way, 3 rotations carry an entropy of 18.5 bits. In contrast, 4 to 9 taps in the baseline prototype carry 28.2 to 63.5 bits, and 4 taps in the enhanced group carry 28.7 bits. For comparison, a standard four to six digit PIN code has an entropy of between 13.3 and 19.9 bits.

Additionally, we also apply the NIST 800-90B test suite [88] to our tap motion dataset⁷. The NIST 800-90B test suite is a widely recognized standard for evaluating the min-entropy of random number generators. It consists of ten distinct and independent tests, each of which calculates a min-entropy value based on a bitstring derived from a given randomness source. The results of applying this test suite to our concatenated bitstring are presented in Table 5.4. To enhance readability, in line with [108], we report the estimated min-entropy in terms of “entropy per bit” (*i.e.*, a value between zero and one).

TABLE 5.4: NIST 800-90B statistical test results for taps.

Test	Entropy Per Bit
Most Common	0.96
Collision	0.79
Markov	0.98
Compression	0.65
Multi-Most Common	0.97
Lag Prediction	0.96
Multi-Markov	0.99
LZ78Y	0.97
t-tuple	0.90
LRS	0.97

Typically, previous research has used an entropy threshold of 0.9 per bit as a criterion to determine whether a random number generator passes a specific min-entropy test [109–112] (Some studies have even adopted a more lenient standard, such as 0.7 [108]). Based on these criteria, our dataset successfully *passes* most of the tests (*i.e.*, 8 out of 10). As a comparison, the urand dataset from Unix-like operating systems can pass 9 out of 10 tests [108].

However, it is important to note that the results in the table are only provided as a reference. Ideally, the NIST 800-90B test suite requires a bitstring of at least 1000 kbits to generate a fully valid evaluation, whereas our data consists of only 21.6 kbits. As future work, more extensive studies with larger datasets will be needed to further verify the feasibility of our pairing techniques.

⁶Note this is a preliminary estimate, and more rigorous calculations are necessary in real-world implementations.

⁷Again, we do not apply this NIST test on rotation motions since they are initialized by Java’s reliable SecureRandom API.

5.3.4 Efficiency and Usability Scores

Time Efficiency. Next, we estimate the time for users to perform 3 rotations and 4 to 9 taps. Figure 5.4d shows the overall distribution of rotation time in the user study⁸. The average time needed for 3 rotations with the baseline prototype is 14 s (SD = 7.11), while for the enhanced prototype the mean time is 9.27 s (SD = 4.56). For taps, the mean time needed for the baseline is between 3.52 s (SD = 2.6) and 7.92 s (SD = 5.85), while for the enhanced prototype the time is 3.92 s (SD = 3.12) since only 4 taps are required. Given these estimates, we calculate that the average bit rate for rotations ranges from 1.32 (baseline) to 2 (enhanced) bps. For taps, the average bit rate lies between 7.32 (enhanced) and 8.01 (baseline) bps.

SUS Questionnaire Score. The average SUS scores for the baseline and enhanced prototype are 62.8 (SD = 21.2) and 85.4 (SD = 9.5) for rotations, and 70.7 (SD = 17.3) and 55.7 (SD = 20.8) for taps. The enhanced rotation prototype and the baseline tap prototype generally pass the typical benchmark value of 68 for “good usability” [89]. The scores for the other two prototypes are noticeably lower. It is important to note that the SUS questionnaires were completed after an extensive data collection process, far more demanding than performing just 3 rotations or 4 to 9 taps. We expect that users carrying out a more realistic task would report even higher usability scores.

5.3.5 Learning Effect in Tap Prototypes

In the experiment, we counterbalanced the order in which participants used the baseline and enhanced prototypes. Our analysis indicates that there was a strong learning effect when participants used the tap prototypes. In other words, the sequence in which they used the baseline versus enhanced prototype had an effect on their performance. Figure 5.6 shows the accuracy of tap prototypes, *i.e.*, for 12 out of the 24 participants who first experienced the baseline prototype compared to the rest who first interacted with the enhanced prototype.

The performance of the baseline prototype for participants who started with the baseline provides an insight into the real-world experience of a first-time user attempting random taps (blue lines in Figure 5.6a). Under these conditions, it takes 6 and 10 taps to defend against non-knowledgeable and knowledgeable impersonation attacks, respectively. Conversely, participants who started with the enhanced prototype showed significantly better performance when they later used the baseline (blue lines in Figure 5.6b). With

⁸In both prototypes, we allocated a 10-second window for each rotation instruction. However, the time reflected here is the actual duration taken by participants to complete the rotations, which is often much shorter than the allocated window. Thus, future design can consider reducing the window.

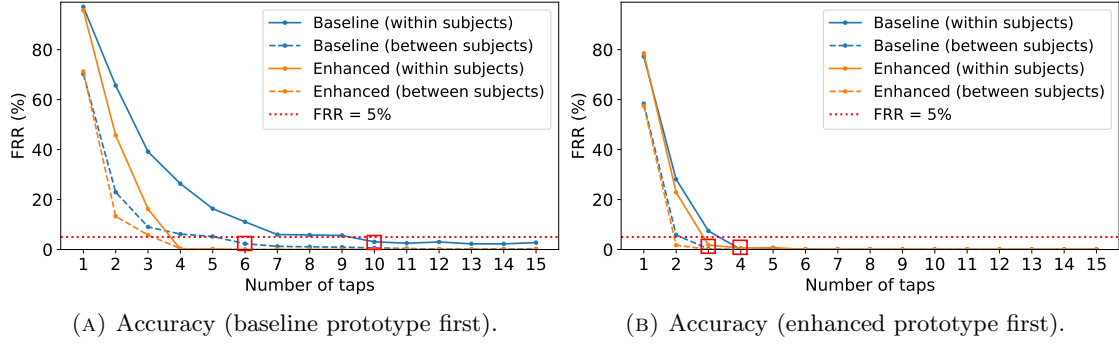


FIGURE 5.6: Learning effect in tap prototypes. Here we split the results from Figure 5.5c, depending on which prototype participants used first. The red squares indicate the minimum number of taps for accurate pairing using the baseline prototype against adversaries of various capabilities (blue lines).

the baseline prototype they required only 3 and 4 taps to defend against both types of attacks. This was the case even though participants in the baseline prototype were not provided with any guidance and were simply asked to tap based on their intuition.

Our interpretation of this learning effect is that the enhanced prototype serves as an effective “tutorial” that encourages users to behave more randomly. The benefits of this training persist even when the guidance is withdrawn. Importantly, this learning effect was achieved in a very short time span—around five minutes for all participants in the experiment—which further underscores the efficiency of the feedback provided by the enhanced prototype.

5.3.6 User Perspectives

To gain a more comprehensive understanding of user attitudes toward our pairing techniques, we carry out a thematic analysis of interviews, revealing three main themes that stand out as crucial factors for establishing IMD pairing through bodily motions.

Physical Comfort and Accessibility. The level of physical demand is highlighted as a key consideration in pairing. Participants who preferred tap over rotation generally felt that tapping was easier and more convenient. In contrast, rotations were seen as more physically demanding, potentially leading to dizziness and posing challenges for elderly individuals with mobility issues. This focus on physical demand was also reflected in their preferences among the rotation prototypes—participants were less fond of the baseline design, which requires precise body alignment to specific degrees. This is more difficult to execute and could lead to more dizziness compared to the enhanced prototype. Overall, two out of the 24 participants mentioned experiencing mild dizziness when using the baseline rotation prototype, but also said that this was due to the repetition of 24 rotations, and that less rotations would alleviate this issue. Furthermore, some

participants expressed concerns about the pairing frequency; nevertheless, they indicated that less frequent pairings, such as once a week, were manageable.

Cognitive Load. In addition to physical requirements, mental load also emerged as a significant factor for participants. Over half of the participants found rotations to be simpler and more straightforward compared to taps, which they felt were more difficult to execute randomly, more boring, or even weird. Participants also commonly favored the enhanced rotation prototype for its clear interface, and often described it as a game-like experience (using terms such as “enjoyable”, “fun”, and “game”). Moreover, the similar group of participants generally did not like the enhanced tap prototype. They found it stressful to discern whether they were executing the motions correctly, and disliked feeling judged. Specifically, many participants noted their dislike to the negative earcon, describing it as “annoying” and “discouraging”.

Sense of Control and Safety. Another critical factor was the user’s sense of control and safety. Many participants highlighted that they felt in control of their pacemaker using our proposed pairing techniques—even though the term “control” was never explicitly mentioned by us in the discussion. Participants also commonly emphasized the need for more transparency in the pairing process. For instance, one participant (p21) stated, “I think I have the right to know immediately and directly, who can pair my pacemaker, who’s pairing and how it’s going to be paired.” Similarly, a sense of safety was a recurring theme. For example, another participant (p5) mentioned, “Even if my doctor is reliable, I want to know when my pacemaker is controlled by someone.” Recall our interview question comparing Medtronic’s solution (as per in Table 3.2) to our designs: 21 out of 24 participants preferred our approach. The remaining three favored Medtronic’s design, and stated, e.g., “I don’t see why anyone would want to attack me” (p7). Overall, these insights further substantiate our motivation for shifting towards a more “seamful” design approach in pacemaker pairing.

5.3.7 Insights From the Co-Design Workshop

Valuable perspectives were also gathered from the co-design workshop. For the rotation-based pairings, the cardiology nurse highlighted potential challenges for pacemaker patients in performing the rotations, stating “People may get low blood pressure when standing up and rotating, or have muscular neurological problems like Parkinson’s”. As an alternative, she suggested having the patient sit in a swivel chair and be assisted by medical staff to complete the rotation, therefore minimizing issues like dizziness. The pacemaker patient affirmed the practicality of this solution; indeed, when asked to execute the rotations, he opted to sit in a swivel chair rather than stand. Additionally, he

expressed that using visual cues from a smartphone during rotations was “unnatural”, and proposed using audio instructions, like “rotate to the right until I say stop” played via headphones, as a more intuitive alternative.

For the tapping prototypes, both elderly participants found the motion simple and easy to execute. They also confirmed that the gentle taps used in our experiment posed no risks to pacemaker patients. Interestingly, both participants preferred the enhanced prototype. The cardiology nurse appreciated the feedback, saying “It actually tells you if you’re on track”. The pacemaker patient liked the different earcons, mentioning “It’s more helpful to have different sounds, and it encourages me towards the end of pairing”. Notably, the pacemaker patient pointed out a safety guideline: electronic devices should be kept at least six inches away from the pacemaker to avoid possible interference [30]. To adhere to this, we attached the motion sensor to the skin near the heart and performed taps six inches away on the right chest area. Our sensor readings still displayed clear peaks, validating the feasibility of our approach.

Furthermore, we commonly observed participants’ limited awareness of cyber vulnerabilities associated with pacemakers. Throughout the interviews, they expressed confusion about the need for secure pairing. For example, they did not understand the potential threat of a seamless pairing process, and asked e.g., “Why bother doing motions if the devices can connect via Bluetooth”. This observation conforms to previous research [113], stating that patients with cardiac implants generally have a limited grasp of the security threats they may face.

5.4 Discussion and Conclusion

The unique nature of IMDs introduces new challenges in the design of suitable pairing methods. An important decision in the design of interactive systems is whether they are designed to be seamless or seamful [114, 115]. Both alternatives come with trade-offs: seamless designs prioritize simplicity and ease of use, while seamful designs focus on enhancing user awareness and engagement. In the context of IMD pairing, existing work largely leans towards seamless designs. However, we argue that this may inadvertently create vulnerabilities, enabling adversaries to launch attacks silently [35, 37, 39, 43]. Our work explores an alternative approach by considering a more seamful design, necessitating some level of physical interaction—a few rotations or taps—from the user during the pairing process.

We empirically validate the feasibility of our techniques through a study involving 24 participants and a co-design workshop. Overall, we find that the workload required to

bootstrap a secure pairing is feasible, and we estimate that it requires only 3 rotations or between 4 and 9 taps for reliable performance, consuming roughly 10 seconds for rotations and 4-8 seconds for taps. With an FAR of 0 and an FRR below 5%, the risk posed by adversaries is low, and legitimate users will likely experience very few failures.

We calculate that the entropy shared through these pairing motions largely *surpasses* that of a standard four-digit PIN code. Recall that the IMD and the external device first share a short secret through bodily motions. They then run a PAKE algorithm to subsequently exchange a long-term encryption key based on the short secret. The PAKE significantly restricts the time window in which an adversary can launch online brute-force attacks (see Section 3.1.3), and also restricts the number of MITM attempts. Typically, the adversary has a very limited period to guess the secret and usually only one chance for a MITM attack; failing within this period would make recovering the encryption key impossible. As an estimate, a shared secret as strong as a 4-digit PIN code could reduce the adversary's success probability on brute-force attacks to as low as 0.01% [78] (assuming the adversary is limited to guessing only), whereas for our techniques this likelihood is even lower. Therefore, we believe that these motions serve as adequate input for a PAKE. Note that, if required, higher entropy can be achieved by performing more motions.

Our user study also validates that the pairing process is easy to understand, learn, and execute, even in their current simplistic prototype form. Taps can be conducted by either the patient or medical staff. On the other hand, although rotations could theoretically be performed on a wheelchair (or stretcher) with the help of medical staff (as mentioned by the cardiology nurse as well), its practicality has not been validated by us.

For each pairing technique, we implement various prototypes for each technique, including baseline and enhanced designs, and evaluate them in a laboratory setting. We show that targeted adjustments to the user interface can, in some cases, improve the effectiveness of the pairing process. For example, refining the interface in the rotation-based technique led to faster pairing and better user experience, with only a minor impact on accuracy. Conversely, we note that some design choices involve trade-offs. In the case of the tap-based technique, enhancing security features had the unintended consequence of diminishing usability. Nevertheless, the robustness of our proposed pairing techniques is confirmed across various implementations. Moreover, the study suggests that further optimization is possible through more user-centered design.

Overall, we find that bodily motion is a promising approach for seamless pairing with IMDs, which can achieve security levels compatible with modern standards. Note that a more comprehensive discussion of our proposed pairing techniques, including their deployability into existing healthcare infrastructures, relevant user training procedures,

comparison with previous work, resilience to side-channel eavesdroppings, limitations and so on, is given in Chapter 8. This is because this part is also shared with another pairing technique we propose in the next chapter.

Chapter 6

Using Vibration to Pair with IMDs

In this chapter, we present our Study III: investigating the use of vibration signals for secure pairing with IMDs. Vibrations are highly perceivable and have been widely used in smart consumer devices for notification services. Additionally, vibration has the potential to be generated and detected by hardware components that are cheap, widely deployed, and miniaturized. In particular, accelerometers, the primary type of vibration receiver used in previous approaches, are already present in state-of-the-art IMDs for medical purposes [57–59]. Another advantage of using vibration is its limited range of reliable reception. In the IMD context, this implies that if an external device intends to transmit a vibration to an IMD, it must be physically attached to the patient’s skin [38]. If an adversary overpowers the signal with a very strong vibration from a distance, the patient can easily notice this.

Not surprisingly, there are various studies that propose the use of vibration signals for pairing or secret transmission between two devices. This includes previous work designed for IMDs, as discussed in Section 2.2. However, as we will demonstrate, such approaches often face significant deployability challenges that hinder their practical application in real-world IMDs. Our work aims to overcome these limitations and propose novel secure, usable, and deployable pairing methods for IMDs.

Similar to our previously proposed approach based on bodily motions, the pairing process in this chapter involves physical interaction with the patient’s body—specifically, repeatedly attaching the external device to the body—combined with vibration signals during each attachment period, to achieve secure pairing. Likewise, this method could be completed by either the patient or a trusted third-party individual like a doctor or a caregiver.

In this chapter, the data from our user study and software used in our prototypes are currently in a temporary anonymous link: https://osf.io/tk56r/?view_only=2be015b407474ac1a97f1bcca682d991. We plan to publish them properly under a permissive open-source license after we publish our Article III.

6.1 Motivation

Previous work has proposed vibration as an OOB channel for transmitting secrets between two devices that physically contact each other [38, 83, 116–120]. Table 6.1 summarizes their application scenarios and hardware setups. Most are designed for wearables and IoT devices that are not implanted in the human body. As a common setup, the transmitter (such as a smartphone) is equipped with a vibration motor and the receiver contains a sensor to detect the vibrations, such as an accelerometer [38, 83, 116, 117, 119], gyroscope [118] or microphone [120]. In particular, there are two primary types of vibration motors in use [121]: ERM motors generate vibration by spinning an off-center weight and are extensively used due to their simplicity and low cost. Linear Resonant Actuator (LRA) motors, on the other hand, vibrate by oscillating a mass along a linear path, offering more precise control over vibration patterns and are being gradually employed by advanced consumer devices.

TABLE 6.1: Setup for previous vibration-based secret transmission. The application context refers to the intended receiver device.

Technique	Application Context	Motor Type	Receiver Sensor Type	Sampling Rate (Hz)
Vibrate-to-Unlock [116]	RFID tag	ERM or LRA	Accelerometer	Not reported
SYNCVIBE [117]	Wearable	ERM or LRA	Accelerometer	1600
SecureVibe [38]	IMD	ERM or LRA	Accelerometer	3200
VibroComm [118]	IoT device	ERM or LRA	Gyroscope	32000
Ripple [119]	Mobile device	LRA	Accelerometer	1600
Ripple II [120]	Mobile device	LRA	Microphone	48000
Touch-And-Guard [83]	Wristband device	ERM	Accelerometer	250

Prior work predominantly directly embeds the secret within the vibration signal itself [38, 116–120]: the transmitter encodes the secret into vibration using specific modulation methods, and the receiver picks up this vibration with a sensor and decodes the secret. For instance, [38, 41, 117] employ On-off keying (OOK) modulation to encode the secret, transmitting a ‘1’ with a carrier signal and a ‘0’ with no carrier. Another strategy leverages vibration to “amplify” the secret from humans: Wei et al. [83] propose an approach that pairs an IoT device with a wristband device, where the IoT device contains a vibration motor and both devices have an accelerometer. When the user (who wears the wristband) touches the IoT device, the motor emits a vibration that sweeps through a range of frequencies. Contrary to the above methods, the vibration here does not

carry the secret and remains consistent across different sessions. Instead, the secret comes from the (to some extent random) resonant properties of the user’s hand-arm area, which can be derived from the accelerometer readings.

However, we argue that most work (in their current form) is not deployable in existing IMD products because they have stringent requirements on the receiver sensor. Microphones do not exist in IMDs, nor is there any indication they will be included in the foreseeable future; while inertial sensors (*i.e.*, accelerometer and gyroscope) often require sampling rates in several thousands of Hz or higher. While there are no published reports on current IMDs’ inertial sensor sampling rates, we doubt whether these devices have been or will be equipped with such high performance sensors: First, a primary function for which inertial sensors are incorporated into IMDs—patient routine activity recognition [57–59]—only takes at most a few hundred Hz to gain high accuracy [122, 123]. Second, other technologies that fulfill specific medical purposes for IMDs typically use sampling rates of up to one thousand Hz [71, 72, 124]. Third, the power consumption of inertial sensors often rises exponentially as the sampling rate increases [122], while IMDs have constrained batteries. Future studies could certainly explore if prior work remains effective at reduced sensor sampling rates such as a few hundred Hz. Nevertheless, this is likely to significantly impact the performance, because vibration signal demodulation often requires sensor data with high resolution [119].

Overall, we find that only [83] demands a lower sampling rate of 250 Hz. This is because the secret relies on the resonant frequencies of the user’s hand-arm region, which are situated in the low-frequency domain ranging from several to a few hundred Hz [125, 126]. Nonetheless, its practicality was only validated for wristbands but has not been tested in other deployment environments or with different hardware setups. For our work, we start with the approach of [83] in our prototype that simulates the environment of an IMD and demonstrate that it may not be the optimal choice. Consequently, we propose a new and reliable vibration-based technique for pairing with IMDs using accelerometers with the same or potentially lower sampling capabilities.

6.2 Method

This section presents the evaluation of a promising approach proposed by [83] in the IMD context, the design of a new vibration-based pairing technique, the development of signal processing workflow, and the data collection process during our user study.

Other methodologies, such as the general prototype setup (e.g., the sensor, chest strap and human body simulator), as well as the participant demographic and recruitment

process, are detailed in Chapter 3, as they are also shared with another pairing technique proposed in the last chapter.

We inherit our assumptions mentioned in Section 3.2.1: We assume that the IMD is a pacemaker equipped with an accelerometer, and will refer to it as “pacemaker” in the remainder of this chapter. We also assume the external device to be a smartphone; however, as we are not using a real smartphone in our study (not like our Study II. The reason is given in Section 3.2.1), we will still call it “external device” in this chapter.

6.2.1 Pairing Technique

Our pairing process requires the user (patient or doctor) to repeatedly attach the external device to the patient’s body (near the pacemaker’s location) for a few times. For our work, we refer to each repetition of the user as a *cycle*, and the complete pairing process (including several cycles) as a *run*. Each cycle comprises three main steps:

- **Device Attachment.** The user attaches the external device to the body and holds it steadily.
- **Vibration broadcast.** The external device emits a vibration signal for a short period. The signal is always the same, and does not serve as the secret. Both the pacemaker and external device take a measurement of the acceleration. The user releases the external device when the vibration stops.
- **Randomness extraction.** Both devices process the sensed acceleration signal and derive a shared secret from it.

The security of pairing relies on the randomness of the shared secret, which originates from the diverse physiological characteristics of the human body as well as the inherent variability of human behavior [64]. In particular, the latter includes, e.g., the varying position where the external device is attached and the spontaneous manner in which the device is held, such as various levels of grip strength. The vibration signal itself remains constant in each cycle and is *not* a source of randomness. Instead, it serves as a “catalyst” that allows the randomness of body and motion to be reflected in the accelerometer measurements.

6.2.2 Obtaining a Shared Secret From Human

The design of vibration strategy in each cycle—namely, the control of the motor to vibrate at a certain frequency(s) for a certain time frame(s)—is crucial. We first explore

the feasibility of the aforementioned work [83] when applied to the context of IMDs. We replicate the exact same experimental settings using our prototype that simulates the human body environment (as shown in Section 3.2.1): The accelerometer sampling rates of the external device and pacemaker are set as 250 Hz¹. In each cycle, the motor is programmed to sweep between 20 Hz to 125 Hz within 1.75 seconds. During this period, two devices measure the Z-axis² (aligns with the user’s sagittal plane) acceleration data and subsequently generate the frequency spectrum by doing Fast Fourier Transform (FFT). One researcher of our research team performs 100 cycles as a preliminary test. The results are shown in Figure 6.1.

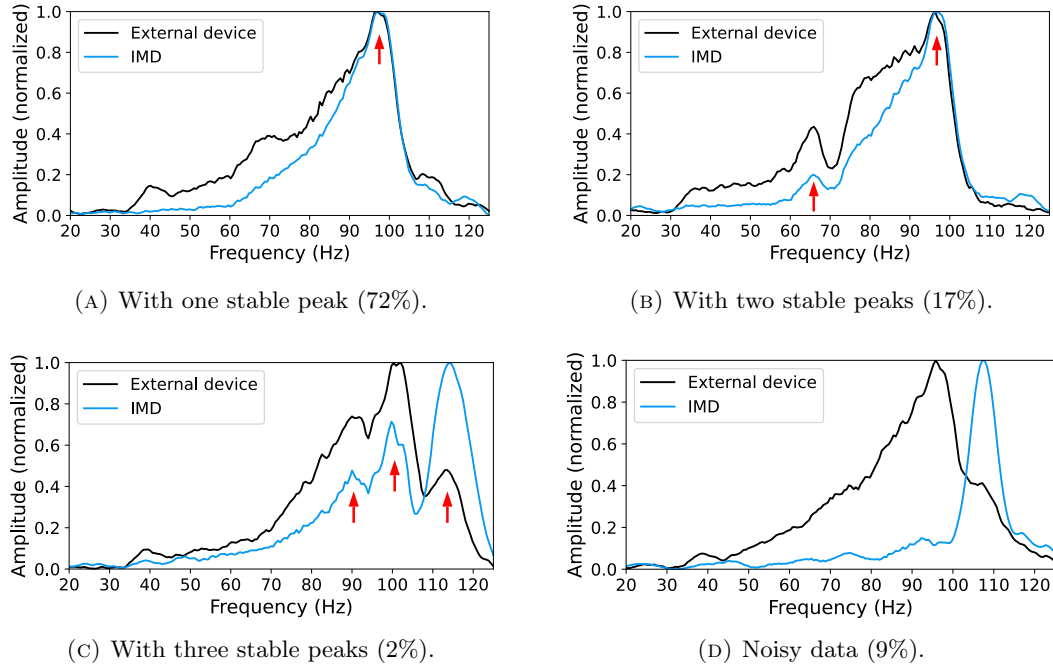


FIGURE 6.1: Performance of swept vibration using our prototype. The peaks shared between devices are marked with red arrows.

For clarity, we normalize the frequency spectrum and manually divide the data into four categories according to the number of peaks shared by both devices (the locations of the resonant frequency peaks were regarded as secrets in [83]). Among all, 72 cycles show one stable peak; 17 cycles have two common peaks while only 2 possess three peaks. Apart from these, in 9 cycles, the data is so noisy that it is impossible to capture any shared peaks. Our results differ significantly from [83] where an average of 4-8 peaks can be obtained per cycle. Additionally, the peaks in [83] are somehow uniformly distributed over the whole 20-125 Hz range, while ours are almost always in the range of 80-110 Hz. Our interpretation for the discrepancy in the performance of this strategy

¹Thus, according to the Nyquist-Shannon sampling theorem, the devices can accurately measure vibrations at frequencies up to 125 Hz.

²Three axes of accelerometer can capture very similar frequency domain information given a vibration. Hence, typically only one of the axes is used to save computational power and energy.

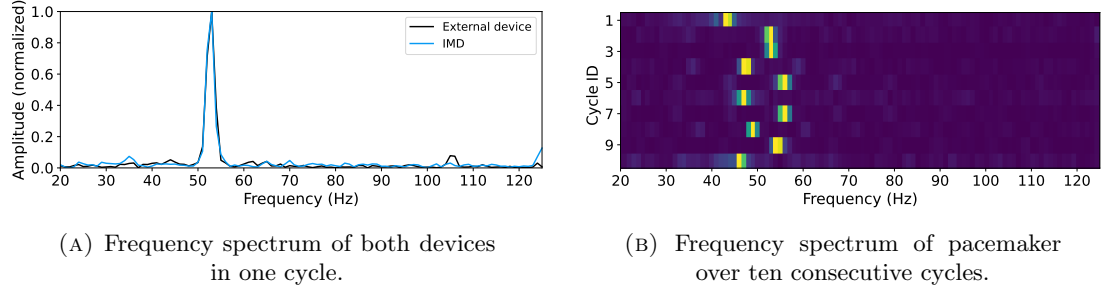


FIGURE 6.2: Performance of constant vibration (50 Hz, 1 s per cycle) by our prototype.

is the presence of the plastic board/shell in our prototype setup, which “masks” the resonant frequencies of the human body. Unfortunately, in the context of IMD pairing, the existence of such components (e.g., a plastic or metal device housing) is inevitable. In practice, we expect that unknown and varied environments may lead to very erratic performance of this approach.

However, we still draw inspiration from the above test. The variance of frequency response among cycles implies the natural randomness inherent in the user attachment motions, given that the physical characteristics of the vibration, the user (*i.e.*, body and hand) and the environment (*i.e.*, prototype) are unchanged.

Intuitively, we want to test if a constant-frequency vibration is a viable option. We program the motor to emit a 50 Hz vibration for 1 s per cycle, and the same researcher executes 100 cycles using our prototype. For each cycle, we collect Z-axis acceleration data from both devices and generate the frequency spectrum using FFT. Figure 6.2a shows an example of the frequency spectrum in one cycle. We observe that two devices are capable of obtaining very similar data, especially for a prominent amplitude peak. Figure 6.2b illustrates the spectrum change of the pacemaker over ten consecutive cycles. Each row in this figure corresponds to a frequency spectrum obtained in one cycle, and the bright spots indicate the prominent peaks on the curve. We observe that the peak locations vary around 50 Hz, suggesting the presence of a degree of randomness. Furthermore, two device capture exactly the same peak locations in 98 out of 100 cycles, yielding a low error rate. These findings indicate that providing an excitement of a constant-frequency vibration, the *prominent peak location* in the frequency domain is a potentially qualified shared entropy source between the pacemaker and the external device, which can be utilized for pairing purposes.

6.2.3 Signal Processing Workflow

Figure 6.3 shows the workflow of our pairing technique. In each cycle, the user holds the external device and attaches it on their chest. During the attachment, the motor

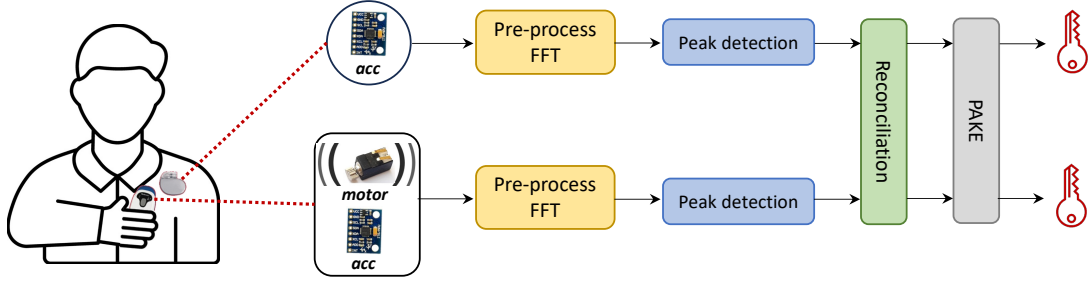


FIGURE 6.3: An overview of our pairing technique.

vibrates and both the pacemaker and the external device measure a pair of Z-axis acceleration data. The two devices first initiate a pre-processing step: To remove the noise of the direct current component, each device subtracts the acceleration data with its mean value. In addition, the frequency transition of a vibration motor is not ideal and instantaneous [38, 117], *i.e.*, when the motor is switched on from standstill or switched off, the generated vibration signal is not amplified or attenuated immediately but with a slow and damped response. This means that the transition parts (*i.e.*, two ends) of a vibration signal segment are often noisy. To address this issue, we apply a Hanning window on the data to suppress the noise at transition frames and highlight the middle part.

Subsequently, each device applies FFT on the acceleration signal to obtain the frequency spectrum. The frequency range of 0 to 20 Hz is then excluded to avoid the effects of noisy motion artifacts like human breathing movements [127]. As mentioned, there is a prominent amplitude peak in the frequency spectrum. In order to detect the location of this peak, each device simply traverses the frequency domain to find the frequency value corresponding to the maximum amplitude. This solution is straightforward and has demonstrated significant robustness in our study, because the prominent peak is often significant enough and overwhelms the other local optima caused by the noise (see Figure 6.2a).

Based on the above procedure, after the user completes a pairing (*i.e.*, a run) by repeating the attachment for several times, each of the two devices will possess a sequence of peak locations. However, these sequences may not be exactly the same due to the measurement noise and human error (e.g., hand wobbles). We design a two-step reconciliation process to resolve this: First, each peak location is encoded into a binary format using Gray code [128]. This coding method ensures minimal bit mismatches as long as the discrepant peak locations are very close on two devices, which is the case of our technique. These binary values are then concatenated in sequence to form a continuous bitstring on each device. Second, we employ a syndrome construction of fuzzy extractor [19, 129] to reconcile any remaining bit differences between the two bitstrings

without revealing the secret itself. This approach is chosen for its high efficiency and low power consumption characteristics, as we will report in Chapter 7. As long as the rate of bit mismatches falls within the error-correcting capability of the chosen fuzzy primitive, the pacemaker and the external device agree on an identical bitstring as a shared secret, which is used as the input of a PAKE to further exchange a cryptographic key (see Section 3.1).

Note that we have intentionally designed the signal processing to be straightforward and lightweight for computation, taking into account the limited resources of the IMDs. We also explored the use of, e.g., signal smoothing techniques, but found that they only slightly decreased mismatches while imposing additional burden on the IMD. We thus omitted such steps.

6.2.4 Data Collection

We manipulate two essential vibration settings, frequency and duration, and measure the effect on pairing performance. We set three levels of vibration motor frequencies to 50 Hz, 75 Hz and 100 Hz, considering that the motors and accelerometers in consumer devices are often limited (recall that an accelerometer with a sampling rate of 250 Hz can only measure vibration frequencies less than 125 Hz). During our preliminary tests, we noticed that due to the inherent instantaneous nature of vibration motors (see Section 6.2.3), the frequency spectrum obtained from vibrations lasting less than 400 ms was often noisy with multiple peaks, leading to high mismatch rates between the two devices. On the other hand, vibrations over one second did not appear to offer additional benefits but harmed time efficiency and impacted user experience. Therefore, we set vibration durations to 400 ms, 700 ms and 1000 ms. The 9 possible vibration frequency/duration combinations are intentionally set to avoid extensive workload for participants in the user study.

An overview of our prototype setup is shown in Figure 6.4a. In the user study, the vibration motor should start to vibrate only when the external device has been well attached to the user's body. To ensure this in our study, we incorporate a buzzer into our prototype that emits a beep sound 0.5 seconds prior to each activation of the motor, serving as a cue for participants to attach the device. This design demonstrated effectiveness during our user study. However, note that this is just a preliminary solution for prototyping purposes. In a real-world set up, the external device can certainly utilize a proximity sensor or a camera to autonomously determine when it has been attached to the body.

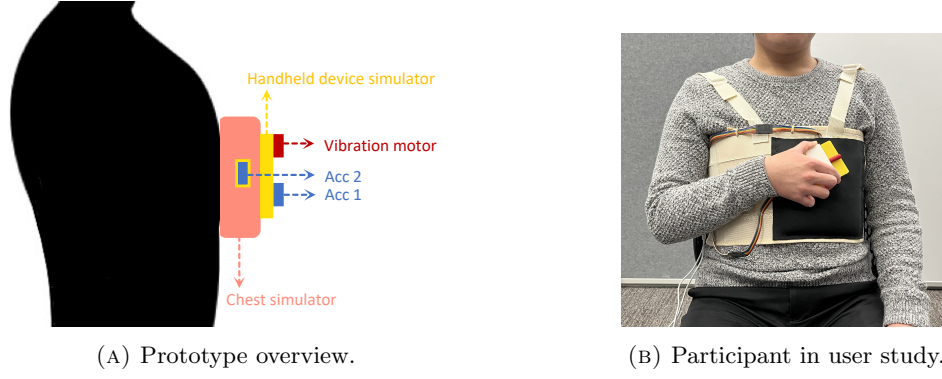


FIGURE 6.4: Experimental setup.

At the start of the user study, participants were instructed to wear our prototype and sit on a chair. Then they need to grasp the external device simulator and repeatedly attach it to the black pocket area of the chest strap, as shown in Figure 6.4b. They were advised to attach the device in a random manner such as to random positions, and (in each cycle) stay attached until the vibration had completely ceased. Before starting the data collection, participants were asked to acquaint themselves with the prototype to understand the pairing process. This introductory process took under a minute for all participants. Subsequently, for each of the 9 vibration conditions, participants were asked to conduct the attachment for 5 consecutive cycles as one run, and complete 4 such runs in total. The order in which participants used different vibration frequencies was counterbalanced, while the order of using different vibration durations was fixed (always from short to long).

At the end of the user study, participants were requested to fill out a standard SUS questionnaire [89] to assess the usability of the pairing method. We then conducted an interview with them to gather further insights. Full details of the questionnaire and the interview are listed in Table 3.1 and Table 3.3, respectively.

6.3 Results

6.3.1 Shared Secrets

Figure 6.5a shows the distribution of all peak locations (*i.e.*, the secret) collected by the pacemaker from 24 participants. The plotted distributions have been slightly smoothed for better visualization. We observe that for a specific vibration frequency, such as 50 Hz, the peak locations range between 30 and 70 Hz and generally approximate a normal distribution centered by the motor's frequency, suggesting a certain degree of randomness from the user. Additionally, it appears that the distribution is slightly flatter

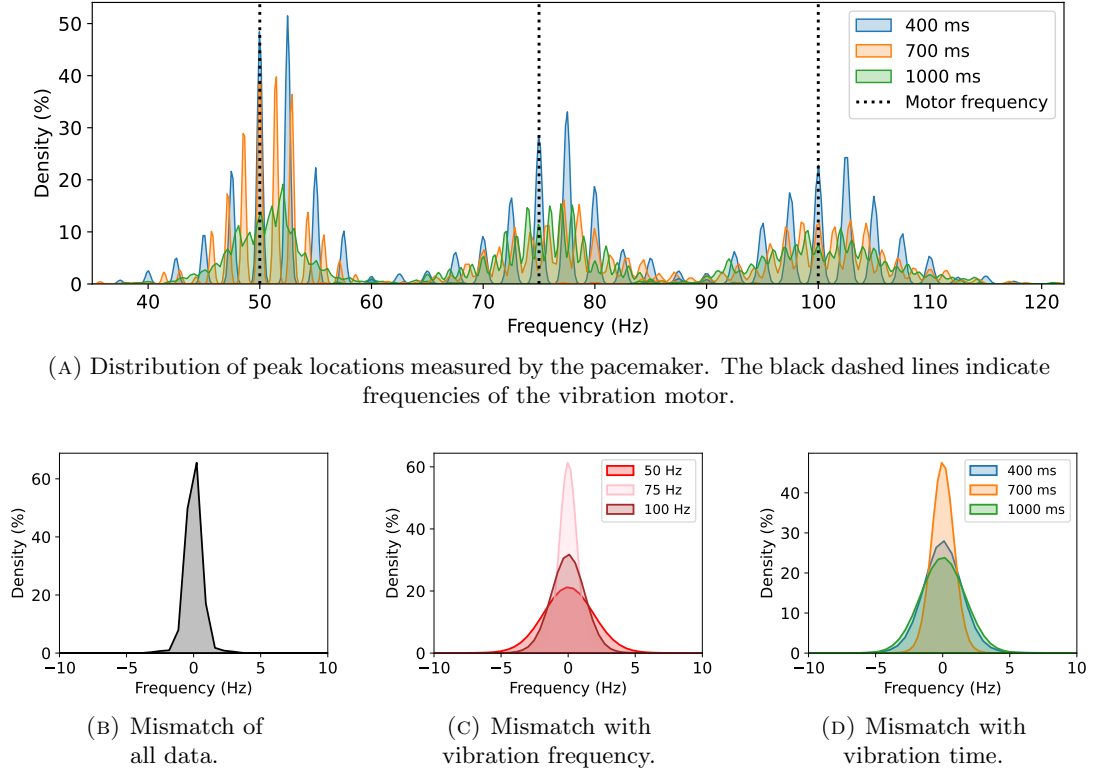


FIGURE 6.5: Distribution of peak location and mismatch among 24 participants.

(thereby the level of randomness increases) with an increase in vibration frequency. This observation is later substantiated by the entropy measurements.

The possible options for peak locations in the frequency domain are not continuous due to the sample-based nature of the time domain acceleration data. Notably, the more time domain samples, the higher the frequency domain resolution. For instance, with an accelerometer sampling rate of 250 Hz, the frequency resolution is 2.5 Hz for a 400 ms measurement window. This indicates that potential peak locations are in discrete intervals such as 47.5 Hz, 50 Hz, 52.5 Hz, and so on. Consequently, we observe more “jagged” curves for shorter vibration durations like 400 ms, and denser curves for longer vibrations such as 1000 ms.

Mismatch is calculated by subtracting peak location values between the pacemaker and the external device, and represents the level of noise and error. The mismatch distribution for our prototype, as illustrated in Figure 6.5b, resembles a normal distribution centered around a mean near zero and with a standard deviation of 2.8 Hz. This implies that user-induced errors and sensor noise are limited in our prototype. Note that this result takes into account situations where participants did not strictly follow our pairing norms during the study. For instance, there were a number of occasions when participants released the external device while it was still vibrating. Such cases were *not* excluded from our dataset as they present a more realistic use scenario; otherwise, we

expect that the mismatch levels would be even lower. On the other hand, Figure 6.5c and Figure 6.5d show that the degree of mismatch does not have a straightforward correlation with either the vibration frequency or the duration.

6.3.2 Pairing Accuracy

For each of the 9 vibration conditions, we build two sets to measure accuracy:

- **Set I** comprises 96 ($= 24 \times 4$) pairs of peak locations, each with a length of five (since we collected five cycles per run). All the pairs in Set I come from legitimate pairings of a pacemaker and an external device.
- **Set II** has 96 pairs of peak locations (the same size as Set I), where each pair is created by randomly mixing data from illegitimate device pairings. An example would be a pair originating from participant A's pacemaker and participant B's external device, or from participant A's pacemaker and external device, but collected in different pairing sessions.

In this study, we did not generate Set III to test resilience to knowledgeable impersonation attacks (as we did for tapping in Study II), due to the limited amount of data (for each of the 9 vibration conditions) collected during the user study. This is a limitation of our work, and future studies should consider conducting larger-scale user studies to test this.

An effective pairing technique should maximize the acceptance of pairs from Set I (*i.e.*, low FRR), while minimizing the acceptance of pairs from Set II (*i.e.*, low FAR). Note that not all five motions are necessarily needed, *i.e.*, we can vary the length of runs ranging from one to five, by truncating the initial elements.

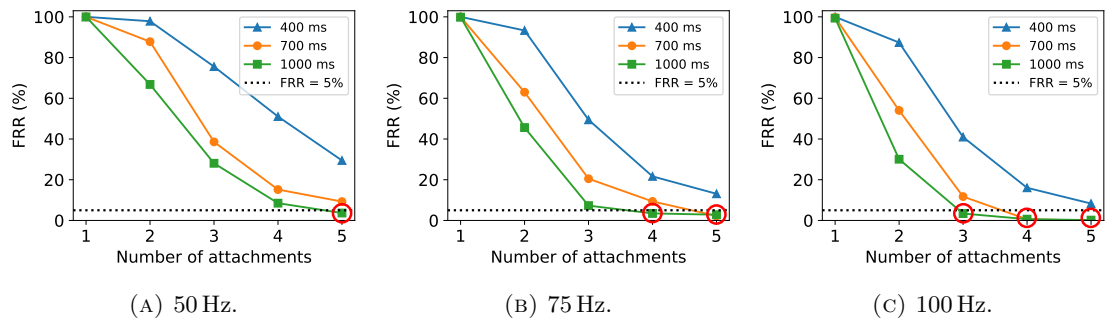


FIGURE 6.6: FRR *vs.* number of attachment motions under different vibration conditions (FAR = 0 in all cases). The red circles indicate configurations with FRR < 5%.

Figure 6.6 shows the accuracy of our prototype across various numbers of attachment motions performed. We consider that an FRR below 5% signifies good usability. As

expected, increasing the number of motions consistently improves the accuracy of the pairing. Moreover, given a specific vibration frequency, longer vibration duration leads to higher accuracy. This is due to the increase of entropy with prolonged vibration, which will be further discussed in Section 6.3.3. An additional observation is the association between vibration frequency and accuracy. With a fixed vibration duration and number of motions, we find that the FRR tends to drop as the vibration frequency rises. This insight requires further investigations.

Overall, the red circles in Figure 6.6 indicate the five out of nine vibration conditions that offer acceptable accuracy levels. For example, a vibration condition of 50 Hz for 1000 ms per cycle requires the user to execute five attachments to achieve pairing with $\text{FAR} = 0$ and $\text{FRR} = 3.7\%$ (see Figure 6.6a). Note that for other vibration conditions, more than five motions are likely to also yield satisfactory accuracy. However, this was not tested by us and would demand more effort from the user, which could harm usability and even safety (e.g., availability in emergencies). As a result, such settings are not considered in the following.

6.3.3 Security Assessment

NIST 800-22 Test Suite Outcome. We refer to previous work [34, 68, 83] to assess the randomness of the secret generated by our technique. The previous accuracy evaluation has led us to consider only the five vibration conditions that demonstrated acceptable levels of accuracy. For each of these conditions, we take the floor of the (fractional) entropy value for that specific setting (see Table 6.3), and extract that number of Least significant bits (LSBs) from each peak location value. These LSBs are then concatenated together following the order in our user study. Subsequently, we combine these bitstrings from all five vibration conditions as a single 8.6 kbits string, and evaluate its randomness using the NIST 800-22 statistical test suite [86].

TABLE 6.2: NIST 800-22 statistical test results for attachment motions.

Test	<i>p</i> -value	Test	<i>p</i> -value
Frequency	0.880	Block Frequency	0.143
Runs	0.111	Longest Runs	0.099
Binary Matrix Rank	0.170	FFT	0.622
Non-overlap Template	0.160	Overlapping Template	0.377
Serial (<i>p</i> -value ₁)	0.638	Linear Complexity	0.716
Serial (<i>p</i> -value ₂)	0.677	Approximate Entropy	0.176
Cumulative Sums (forward)	0.191	Random Excursions	0.216
Cumulative Sums (reverse)	0.259	Random Excursions Var.	0.403

The full results are given in Table 6.2. The outputs of the NIST statistical tests are *p*-values that represent the probability the data is generated by an eligible RNG. If a

p -value is smaller than a threshold (usually 0.01 [34, 68, 83]), the randomness hypothesis is rejected. Table 6.2 shows that all p -values are larger than 0.01 and hence *pass* the NIST tests. Note that we do not conduct Maurer’s Universal Statistical Test, as it requires a minimum data input of 387.8 kbits that is not met in our case. This test was also omitted by previous work [68, 83].

Entropy Analysis. Table 6.3 shows the entropy value contained in each motion across different vibration conditions. Overall, a single motion in our study carries an entropy from 2.61 to 4.48 bits. For a certain vibration frequency, the entropy grows with higher vibration durations. This is because given a fixed sampling rate of the accelerometer, a longer measurement brings about a larger sample size and thus higher frequency domain resolution. This provides more possible peak locations and thus increases the entropy level. Furthermore, for a given vibration duration, the entropy value rises with an increase in vibration frequency. This observation aligns with the findings discussed earlier in Section 6.3.1. We leave the study of this phenomenon to future work. Nevertheless, there is an upper limit to the choice of vibration frequency, often limited by the capability of the motor and accelerometer in practice.

TABLE 6.3: Entropy of each attachment motion (unit is bit).

	50 Hz	75 Hz	100 Hz
400 ms	2.61	3.12	3.87
700 ms	3.03	3.86	4.15
1000 ms	3.15	4.01	4.48

It is noteworthy that some entropy is sacrificed due to the reconciliation process that rectifies mismatches between the two devices. Here, we make a preliminary estimation of the entropy loss: Using the encoding method in [68] on our dataset, the maximum bit mismatch rates (*i.e.*, percentage of different bits between two devices) for our prototype vary between 0.7% and 3.0% for different vibration conditions. This can be addressed by a fuzzy extractor with (31, 29) RS code that has a 3.23% error tolerance [19, 107], potentially leading to an entropy loss of 6.5%³.

Similar to our work in the previous chapter, we apply the NIST 800-90B statistical test suite [88] to our dataset of attachment motions to evaluate the min-entropy level. The results are presented in Table 6.4.

Using the same standard as previous studies [109–112], our dataset *passes* most of the min-entropy tests (*i.e.*, 7 out of 10). This provides a degree of validation of the randomness in our proposed pairing motion. Nevertheless, we emphasize that these results

³This is a preliminary estimate, and more rigorous calculations are necessary in real-world implementations.

TABLE 6.4: NIST 800-90B statistical test results for attachment motions.

Test	Entropy Per Bit
Most Common	0.96
Collision	0.85
Markov	0.98
Compression	0.54
Multi-Most Common	0.94
Lag Prediction	0.95
Multi-Markov	0.95
LZ78Y	0.94
t-tuple	0.88
LRS	0.95

should be viewed as a reference, as the NIST 800-90B test suite ideally requires 1000 kbits of input [88], whereas our dataset contains only 8.6 kbits.

6.3.4 User Perspectives

The average SUS score for our pairing technique is 73.6 (SD = 18.14), which generally passes the typical benchmark value of 68 for “good usability” [89]. It is important to note that the SUS questionnaires were completed after an extensive data collection process including a repetition of 180 attachment motions. We expect that users carrying out a more realistic task would report even higher usability scores.

We gained further insights into usability from the interviews. Over half of the participants (15 out of 24) *explicitly* indicated that our technique was easy to use. For example, one participant (p8) commented “The attachment doesn’t require me to think. This is an advantage. I don’t know what is happening here, but I prefer it as it requires less effort”, and another participant (p13) remarked “It’s easy. You don’t really have to move that much, and you can do it while you’re sitting as well”. Some participants expressed their preference for the vibrational feedback. One participant (p1) said “The vibration is good feedback, and I don’t have to visually see anything”, and one participant (p21) mentioned, “I feel more in control”, while another participant (p22) noted “The process is like listen to my heart”. Additionally, some participants conveyed that they found the pairing process to be enjoyable and fun. For example, three participants described the vibration as a hand massage and two compared the pairing activity to using a stethoscope.

The majority of participants (18 out of 24) experienced no discomfort during the study. Nonetheless, the rest six people did report some discomfort at the end of the study. Four participants noted that the intensity of the vibrations was excessive, e.g., one participant (p4) stated, “I feel like my entire chest is vibrating, and I don’t like the feeling”. This

concern might be resolved by selecting a vibration motor with lower amplitude. Additionally, four participants reported feeling fatigued after the data collection process, but also noted this was due to the repetition of 180 attachments and that less motions will alleviate this issue. Furthermore, one participant (p3) criticized the prototype design and mentioned that the external device simulator was too big. We leave the refinement of our prototype as future work.

6.3.5 Optimal Configurations

As a reminder, the attachment motions aim to exchange a short secret between two devices, which then serves as the basis for executing a PAKE. Based on our analysis so far, we summarize all pairing configurations that: (i) exhibits high accuracy with zero FAR and FRR under 5%, and (ii) generates a level of entropy *surpassing* a standard four-digit PIN code (with an entropy of 13.3 bits), which is commonly used in pairing of Bluetooth technologies and other security systems [79]. All viable settings that meet these requirements (with minimum required number of motions) are shown in Table 6.5. The presented entropy accounts for the loss incurred during the mismatch reconciliation process. Moreover, the time values include both the vibration duration and an additional “preparation time”, which refers to the interval necessary for a user to detach and then re-attach the external device to their body; in our study, this interval was 0.5 seconds. In summary, we find that with a vibration configuration set at 100 Hz and 700 ms, a user can carry out 4 attachment motions to enable the exchange of a secret with (FAR, FRR)=(0, 0.6%) and entropy of 15.5 bits. This process can be completed in a mere 4.8 seconds, corresponding to a data throughput of 3.2 bps.

TABLE 6.5: Summary of well-performing pairing configurations.

Vibration Condition	# motion	(FAR, FRR)	Entropy (bit)	Time (s)
50 Hz, 1000 ms	5	(0, 3.7%)	14.7	7.5
75 Hz, 700 ms	5	(0, 2.2%)	18.0	6
75 Hz, 1000 ms	4	(0, 3.5%)	15.0	6
100 Hz, 700 ms	4	(0, 0.6%)	15.5	4.8
100 Hz, 1000 ms	4	(0, 0.6%)	16.8	6

6.4 Discussion and Conclusion

The use of vibration as an OOB communication channel for pairing (*i.e.*, key exchange) has gained attention, largely due to the enhanced perceptual experience it provides to users, which is desirable for security applications. However, existing work in this domain (including [38] that focuses on IMDs) encountered practical deployment challenges

when considering existing IMDs. These challenges stem from the high sensor sampling requirements or erratic response to environmental changes, as we have verified in our study.

Our work introduces a new and reliable approach, which only requires a low sampling rate accelerometer and relies on the natural randomness inherent in human behavior for secure pairing. We empirically validate the feasibility of our technique through a study involving 24 participants. Overall, we find that the workload required to bootstrap a secure pairing is feasible, and we estimate that it requires the user to attach a device to the body only 4 times, with the whole process taking roughly 5 s. With an FAR of 0 and an FRR of 0.6%, the risk posed by adversaries is low, and legitimate users will likely experience very few failures.

The entropy shared through the pairing motions exceeds that of a standard four-digit PIN code which is prevalent in security applications [79]. As mentioned in Section 3.1, a proper implementation of PAKE restricts the number of online MITM attempts. As an estimate, 4 motions with 15.5 bits entropy reduce the adversary’s success probability on online brute-force attacks to a value as low as 0.002% [78] (assuming the adversary is limited to guessing only). Therefore, we believe these motions serve as adequate input for a PAKE. It is noteworthy that if needed, higher entropy can be easily achieved by performing more motions.

Our user study confirmed the high usability of our pairing method. Participants found it straightforward to understand, learn, and perform. The process of attaching the device is very intuitive, like using a stethoscope as described by the participants. As an example, one participant (p2) displayed a level of ease and nearly fell asleep during our intensive data collection, yet they continued to perform the motions properly, even with their head dropping and eyes closed. Additionally, our technique brings about certain entertainment to users, being both relaxing and enjoyable (such as described as hand massage). This aspect could be advantageous in certain therapeutic treatments, where physical interaction can enhance memory, concentration, and mental health [130]. Moreover, it is worth noting that for patients who are unable (e.g., due to disabilities or unconsciousness in emergencies) or unwilling to execute the motions, our pairing allows medical practitioners or caregivers (who have received appropriate training) to execute the motions on the patient’s behalf.

6.4.1 Comparison with Prior Vibration-Based Work

Our pairing technique significantly relaxes the demands on the IMD’s sampling capability. We utilize an accelerometer operating at 250 Hz, in contrast to previous work

that often relies on sampling rates of several thousand Hz or more. In particular, the sampling rate can be further decreased by using lower vibration frequencies. For example, with a 50 Hz vibration, the frequency domain peaks cluster between 30 and 70 Hz (see Figure 6.5a), indicating that an accelerometer with a 140 Hz maximum is adequate. However, lowering the sampling rate will diminish the resolution in the frequency domain and thus decrease the entropy per motion, and requires extended sampling durations or more motions to compensate for the reduction in entropy.

Another advantage of our pairing technique is that it forces multiple physical interactions with the patient, *i.e.*, repeatedly attaching the external device to the patient's body. Such bodily motions can add an additional layer of perception beyond the vibration signals themselves. In contrast, most prior work requires just a single initial touch. Moreover, conventional approaches typically try to avoid user-generated noise. For instance, the user needs to ensure stable contact between devices during data transmission. Conversely, our method harnesses user noise and benefits from it as a source of entropy. Indeed, our dataset includes many instances with significant user noise/error, like when a participant releases the external device before the vibration completely stops. In such scenarios, the IMD only captures a portion of the vibration within its measurement window. Despite this, our technique maintains high reliability.

Furthermore, previous work that encodes secrets into vibrations often demands precise time synchronization in milliseconds between devices, which itself is a challenging task for resource-constrained devices [131]. In contrast, our approach allows for more lenient synchronization—as long as the two devices capture similar vibration signals within most of their measurement windows, the peak locations effectively match. This aspect greatly simplifies the synchronization requirements, enhancing the feasibility of our technique for IMDs or other devices with limited resources.

Notably, our data throughput is significantly lower than [38, 117–119, 119], and is comparable with [41, 83]. Considering the scenario of transmitting a four-digit PIN code for use in a PAKE, previous work [38, 117–119, 119] only needs 0.0004 to 0.665 s, which is much faster than the 4.8 seconds required by our method. However, this rapid transmission, while advantageous in many daily applications, may not be suitable for IMD pairing contexts. In such contexts, the vibration serves not only for secret exchange but also as a crucial cue for patients to be aware of the pairing process. For instance, vibrations lasting less than a second might be too fleeting for patients to notice and react accordingly [132], *e.g.*, stopping an adversary's actions, thus raising significant security concerns. In contrast, we argue that a duration of 4.8 s strikes a balance: it is long enough to be noticeable, yet short enough to maintain usability and safety in emergencies.

6.4.2 Potential Health Implications with Vibrations

Our proposed pairing technique incorporates the use of vibrations, a feature that naturally raises concerns regarding the potential long-term health impacts on patients. However, it is important to note that current research indicates that *only* long-term and excessive exposure to vibrations (e.g., due to specific occupations) is linked to adverse effects on mental and physical health [133, 134]. In contrast, our method involves brief vibrational interactions, which last less than five seconds and may not occur every day. This limited exposure could reduce the likelihood of the negative health consequences.

Despite this, we acknowledge the critical importance of thoroughly understanding any health implications associated with our approach, in particular in the delicate context of patient care. To this end, future collaborations with medical domain experts (such as physicians and representatives from medical companies) would be necessary to facilitate more comprehensive evaluations, and ensure that any health risks are meticulously assessed and addressed.

Overall, we find that vibration can be utilized to achieve secure, usable and deployable pairing with IMDs. We will provide a more thorough discussion of our pairing technique, such as their deployability into existing healthcare infrastructures, relevant user training procedures, comparison with previous work, resilience to side-channel eavesdroppings, limitations and so on, in Chapter 8. These aspects are also shared with the pairing technique we propose in the previous chapter.

Chapter 7

Comparing Fuzzy Cryptographic Primitives on Resource Constrained Devices

Deployability is crucial for pairing methods in IMDs. First, IMDs are equipped with limited hardware components, primarily for basic medical functions and wireless communications. It is challenging to add additional hardware for security purposes, as these components would inevitably compete for space with the battery. Second, IMDs operate on limited battery power, where every Joule is vital. Depleted batteries necessitate surgical replacement, posing risks to patients. Third, IMDs are life-critical devices, prioritizing medical functions above all else. Any security protocols must not interfere with these primary functions and must be thoroughly tested for reliability in such devices.

In this chapter, we present our Study IV: studying the deployability of fuzzy cryptographic primitives in resource-constrained IMDs. The source code of the algorithms involved in this chapter, and their corresponding documentations, are organized under a GitHub link with an open-source license: <https://github.com/MrZMN/FuzzyKey>. In particular, our implemented decoding methods for RS code were based in part on the source code of Simon Rockliff [135]. We thank Simon for his work, and acknowledge him in the comments of relevant programs as required.

7.1 Motivation

We aim to ensure that the complete pairing protocol we propose in Section 3.1 is deployable in existing IMDs. Here, we assume that an IMD has similar capabilities to a

Medtronic Azure pacemaker equipped with a 32-bit Cortex-M microcontroller (as presented in Chapter 4), or even lower capabilities, such as 16-bit microcontrollers with more limited computation power and memory.

Our proposed pairing protocol consists of three main parts:

- (1) **Exchange of a Short Secret.** Based on our proposed OOB channels, *i.e.*, bodily motions or vibrations, a short secret is exchanged between the IMD and the external device (*i.e.*, sc and sc' in Figure 3.1). However, this exchanged secret is not always identical but at best rather similar due to inherent noise.
- (2) **Fuzzy Cryptographic Primitive.** Using a fuzzy primitive, an identical and ephemeral key (*i.e.*, STK and STK' in Figure 3.1) is further exchanged based on the previously shared secret.
- (3) **Password-Authenticated Key Exchange.** Finally, a PAKE is executed to exchange a long-term session key (*i.e.*, LTK and LTK' in Figure 3.1) based on STK .

The deployability of part (1) has been verified through our prototype development and human validation processes. For full details of our implementations, we refer readers to Chapter 3, Chapter 5 and Chapter 6. The deployability of part (3) has been well studied and verified in previous work on 32-bit Cortex-M microcontrollers [68]. However, based on our literature review, the deployability of part (2)—using fuzzy cryptographic primitives on resource-constrained devices—has not been fully studied, tested, and compared. Therefore, in this chapter, we thoroughly investigate such algorithms, implement them on resource-constrained devices, compare their performance, and select the ones that best suit the use case of IMDs.

A comprehensive introduction to the six fuzzy cryptographic primitives we evaluate has been provided in Section 2.4. Like that part, the content presented in this chapter is based on our published work in [19].

7.2 Method

All the algorithms were implemented and evaluated on 16-bit microcontrollers. We did not use 32-bit microcontrollers because this study was conducted before our investigation of Medtronic's Azure model (see Chapter 4). Nevertheless, if the fuzzy primitives can operate efficiently on our 16-bit systems, we believe that they can also be easily deployed on more powerful 32-bit systems.

The notations used in this chapter are consistent with those when we introduced the fuzzy cryptographic primitives in Section 2.4. For instance, we refer to the IMD and the external device as TX and RX (or RX and TX¹), and we denote the inputs for each fuzzy primitive, i.e., the data originating from sensor measurements with some potential noise, as sc and sc' .

In our pairing methods proposed in Chapter 5 and Chapter 6, we assumed that the inputs of the fuzzy primitives are in bit string format, which are materials suitable for Hamming metric fuzzy primitives. However, they can also be converted into set formats. Therefore, in this chapter, we evaluate both Hamming and set metric methods.

To ensure a systematic and fair comparison between various fuzzy primitives under different metrics, we assume that all methods aim to exchange a 128-bit cryptographic key based on sc and sc' . However, it is important to note that, in practice, fuzzy primitives do not necessarily need to establish such a robust key as STK in our protocol in Figure 3.1. This assumption is made solely for comparison purposes, and the comparison results are expected to remain consistent even if the length of the output (i.e., the resulting key) varies.

7.2.1 Design Security and Parameter Selection

Attack Vectors. We define the concrete attack vectors on fuzzy primitives based on the adversary capabilities we outlined in Section 2.1.1 and Section 3.1.3. We consider a strong adversary who knows all details about the used fuzzy primitives and has full access to the communication channel between the two devices. The adversary can perform *passive attacks* by eavesdropping on the communication and exploiting information leakage from it. For example, if sc (or sc') in a fuzzy commitment is low-entropy, the adversary can statistically analyze its distribution and may compromise the security [136]. On the other hand, the adversary can also carry out *active attacks*, i.e., act as MITM or replay old sessions. Finally, the adversary can also observe and exploit secret-dependent timing leakage, e.g., the precise time between two protocol messages, both in passive and active attacks.

Countermeasures Against Passive Attacks. For fuzzy commitment and fuzzy vault, the security is reflected in how well the key is concealed. Concretely, because we omitted the binding process, the underlying security of the fuzzy commitment only depends on the entropy of the bit strings sc and sc' as they are used to conceal the key while being transported. The security of fuzzy vault relies on the fact that the adversary cannot distinguish between valid and chaff points, and hence is unable to reconstruct

¹The IMD should be either the TX or RX, whichever has the lower computation load.

$p()$. For a fuzzy vault scheme with parameters g , b and d (see Figure 2.2), the adversary would need an average of $\binom{g+b}{(g+d)/2} / \binom{g}{(g+d)/2}$ attempts to reveal $p()$ (assuming that Berlekamp-Welch decoding is used). Therefore, the number of chaff points needs to be sufficiently large to resist brute-force attacks.

As for fuzzy extractors, the security depends on the entropy of the shared secret sc and sc' themselves, because the key is directly extracted from them using a strong extractor. Because of the leakage of helper data, there will be an amount of entropy loss on sc in each construction. For code-offset and syndrome constructions, the entropy loss is $(n - k) \cdot f$, where n, k come from the underlying (n, k, t) ECC, and f is the number of bits constituting each symbol. The entropy loss is $t \cdot \log_2 u$ for the improved Juels–Sudan construction, and is $t \cdot \log_2(u + 1)$ for the Pinsketch construction, where u is the universe size and t is the maximum set difference between sc and sc' . Note that the above represent worst case entropy loss values [1]. Some of these values were also proven to be overly pessimistic [137]. To ensure security, we regard the worst-case values as the actual entropy loss in this thesis. Suppose we focus on 128-bit key exchange, as long as the remaining entropy of sc (the agreed sc before input to the strong extractor) is ≥ 128 bit, the fuzzy extractor can be considered secure.

Countermeasures Against Active and Timing Attacks. In our protocol, replay attacks are by design prevented due to the randomness inherent in the bodily motions and vibration signals among different pairing sessions. Other active attacks (such as guessing-based ones) require the adversary to break the fuzzy primitive “online” within a single protocol session [68, 138], and they can thus be prevented by generating an sc with substantially high entropy. Timing attacks can be generically prevented using constant-time implementation techniques, which we further discuss in Section 7.2.2.

Assumptions on Fuzzy Primitive Input. The inputs of fuzzy primitives (*i.e.*, sc and sc') are extracted from sensor measurements by two devices. There are several factors that affect the similarity of sc and sc' , *e.g.*, the level of noise and measurement accuracy of the sensor. To evaluate and compare different fuzzy primitives, it is necessary to consider pre-defined mismatch rates (*i.e.*, percentage of different bits/set elements) between sc and sc' , which reflect the characteristics of different kinds of scenarios. In this thesis, we consider three mismatch thresholds of 2%, 5% and 10%².

We also note that unlike BCH codes, RS codes are multi-bit-symbol based. Thus, for RS code variants, the above thresholds indicate the percentage of different symbols rather than bits. Additionally, one should note that the average bit error rate (*i.e.*, the

²In Chapter 5 and Chapter 6, our evaluation results indicate that the mismatch rate between sc and sc' is lower than 5% (or even 2% in certain cases) for our proposed pairing techniques. However, it is important to note that in other scenarios (*e.g.*, when used by other patients), there might be slightly higher mismatch rates.

possibility that each bit differs for two bitstrings) may be more broadly used on the Hamming metric in other application scenarios, thus, we also provide this information for each variant in Table 7.2 (with a maximum tolerable failure rate of 10^{-6}). This maximum average bit rate that can be tolerated needs to be considered when selecting the most appropriate error correcting code.

Here, we assume that sc and sc' are ℓ -bit strings that are random and uniformly distributed for Hamming metric methods, or sets containing s distinct elements that are uniformly distributed in a universe U for set metric methods. This assumption is *only* made for fairly comparing different fuzzy primitives; both fuzzy commitment and fuzzy vault naturally require the input to be uniformly randomly distributed to ensure security³. Additionally, if the fuzzy primitive inputs are not uniformly distributed, it is hard to quantify the entropy level of the sc and establish a unified mismatch rate threshold. Under this assumption, the initial entropy of the sc is ℓ for Hamming metric fuzzy extractors, while for set metric fuzzy extractors, it is $\log_2\binom{u}{s}$ with u the size of U .

Parameter Selection. The mismatch rate between sc and sc' directly determines the error tolerance requirement of the fuzzy primitives. For Hamming metric methods, the error tolerance is the same as that of the underlying ECC (*i.e.*, t/n for an (n, k, t) code). For example, (50, 44, 1) and (20, 15, 1) BCH codes are suitable when the maximum mismatch rate is 2% and 5%, respectively. However, RS codes cannot provide exact 2%, 5% and 10% error tolerance because of their inherent structure. Therefore, we selected several RS constructions with error tolerance within 2% + the pre-defined mismatch rate thresholds. For example, the error tolerance of a (31, 29, 1) RS code is 3.23%. As mentioned in Section 2.4.1, all codes stay within the field $GF(2^8)$. For set metric methods with (u, s, t) structure, where u is the universe size, s the set size and t the maximum tolerable set difference between sc and sc' , the error tolerance is $t/2s$.

For each fuzzy primitive, there can be multiple feasible parameter choices under the same mismatch rate, e.g., using different configurations of ECCs may achieve the same error tolerance. The difference between them is that the repetition count might be different: Assume the total repetition count is r and the number of secret bits distributed in each iteration is i , we need to ensure that $r \cdot i \geq 128$ to achieve 128-bit security. This way, RX concatenates the secret bits it receives in each iteration to form the 128-bit key. The security of subsequent/parallel execution based on linear codes has been proven in [137]. Note that i is the length of the key distributed per iteration for fuzzy commitment and fuzzy vault, but the remaining entropy of sc for fuzzy extractors. Consider a fuzzy commitment that is based on (50, 44, 1) and (200, 168, 4) BCH codes (both handle mismatch $\leq 2\%$) as an example. In order to distribute a 128-bit key, the former variant

³Note that fuzzy extractors can still be securely used even if the inputs are not uniformly distributed.

needs to be executed three times ($3 \cdot 44 > 128$), while the latter only needs to be executed once. Although more iterations may be required, small parameter choices (e.g., an ECC with a small block size) almost always mean less computation and hence less energy consumption. Therefore, we test different variants under the same mismatch threshold. A variant with larger parameter choice is considered only if it reduces the number of required iterations. Besides, under each mismatch rate, the variants used by different fuzzy primitives in each metric are the same, thus help with the performance comparison.

Note that for Hamming metric methods, the number of feasible variants depends on the number of underlying ECCs that achieve the pre-defined mismatch thresholds. We give all feasible variants for Hamming metric in Table 7.2. For set metric methods, there are more possible (u, s, t) variants because (i) the universe size u can vary depending on how a sc is converted to a set and (ii) multiple s and t combinations can achieve the same error tolerance. In this thesis, we use $u = 255$, which is the maximum universe size for the Pinksch construction on $GF(2^8)$. We define three variants $(255, 50, 2)$, $(255, 20, 2)$ and $(255, 10, 2)$ for 2%, 5%, and 10% mismatch rate thresholds. These variants are provided for reference only, and one could devise more appropriate variants for set metrics with specific mappings from sc to set. Finally, the fuzzy vault construction over $GF(2^8)$ is insecure, because the number of chaff points is $\leq 2^8$. However, for our performance evaluation in Section 7.3, we limit ourselves to $GF(2^8)$, and note that the system can be easily extended to larger fields (e.g., $GF(2^{16})$).

7.2.2 Implementation

We implemented, ran, and measured all algorithms on a TI MSP430FR5969 Launch-Pad development board [139]. This board comprises a 16-bit microprocessor with 2kB volatile SRAM and 64kB permanent FRAM, which is representative for low-power body area network devices (including e.g., IMDs). We also alternatively used an MSP430FR5994 development board [140] with 8kB SRAM for certain variants that require more resources, and indicate this in Table 7.2. Note that the capability of either microcontroller is significantly lower than the SoC used by Medtronic’s Azure pacemaker, as demonstrated in Chapter 4. For development, we used TI’s Code Composer Studio as it provides integrated functionality for on-device energy consumption measurement.

Implementation of Strong Extractor. Considering many embedded microcontrollers (including the MSP430 used in this thesis) feature a hardware AES accelerator, we opted to use a block cipher-based hash function, with AES as the underlying cipher. We selected the Hirose construction [2] for the strong extractor in our implementation.

Hirose is a double-block-length hash with Merkle–Damgård structure. We measured the average energy consumption of each invocation of the Hirose compression function on MSP430FR5969 to be 1.42 μJ . Depending on the availability of a fast hardware/software implementation, other hash functions such as SHA256 can be used instead of Hirose.

Software Development and Energy Measurement. We implemented all algorithms in plain C and mainly relied on standard C libraries so that our implementation can be easily ported to other platforms. For random number generation and hardware-accelerated AES, we used TI’s driver APIs. The tested average energy consumption of generating 16 bytes when using TI’s random number generator API is 2.5 μJ . Certain components can be implemented in different ways. For example, there are a variety of algorithms for ECCs. We chose commonly used, efficient algorithms: for BCH and RS encoding, we used standard cyclic code encoding, and for decoding we used the Berlekamp decoding method [107, 135]. For polynomial reconstruction, we used the Berlekamp-Welch algorithm.

We carried out the energy consumption measurement using TI **energyTrace** tool. This functionality allows to take accurate on-device energy measurement from the Code Composer Studio IDE. For each measurement, we averaged the energy consumption value over 100 executions of the respective algorithm. To test the error correction ability of the fuzzy primitives, we artificially added the maximum tolerable number of mismatches on *sc* in the code, and then measured the corresponding energy consumption. We used TI **Ultra Low Power Advisor** tool to refactor our code and minimize energy consumption. Overall, we found that these optimizations reduced the energy consumption $\leq 10\%$.

Estimation of Communication Energy Cost. The energy consumption of a protocol between multiple devices comprises two components: (i) the energy consumption of computations; and (ii) the energy consumption of wireless communication. In this thesis, we only measure the energy consumption of computation, and model the cost of wireless communication based on the number of bits to be transmitted and received. In particular, in Section 7.3, we use the experimental results of [141] for a TelosB [142], a wireless sensor node based on a 16-bit MSP430 microcontroller and a CC2420 transceiver to illustrate the impact of communication overhead on overall energy consumption. Their results show that for 75 kbps data rate and -5 dBm transmit power, the average energy required to transmit one bit of effective data is 0.72 μJ , and the energy required to receive this bit is 0.81 μJ .

We acknowledge that a simplistic “energy-per-bit” model may be inadequate e.g., when using packet-based protocols such as BLE⁴, where the constant overheads due to the frame structure and other steps (e.g., wake-up and preparation) can be substantial.

⁴As mentioned, BLE is already being used in commercial IMDs e.g., Medtronic Azure pacemakers [8].

Therefore, we also provide the number of payload bits for each variant in Table 7.2, which can be fed into a more appropriate energy consumption model for a specific wireless protocol (e.g., informed by measurements as reported in [143]). We note that many widely used protocols support payloads large enough to accommodate all our variants in one packet (e.g., 246 bytes for BLE). This minimizes the impact of the frame structure, thus, when considering such protocols, the different implementations can be compared purely based on their computational energy cost.

Defenses Against Timing Side Channels. We implemented countermeasures against timing-based side-channel attacks on the best-performing variants (*i.e.*, fuzzy primitives with lowest total energy consumption under different mismatch thresholds, see Table 7.4). We found that non-constant execution time mainly arises in the ECC encoding/decoding processes through various conditional branches depending on a value being negative. To address this, we replaced all such conditional branches with Boolean operators, and used other constant-time implementation techniques, such as constant-time modulo reduction (based on Barrett reduction) and constant-time sorting.

TABLE 7.1: Effects of timing side-channel defenses on timing leakage (measured by Welch’s t-test) and energy consumption.

fuzzy primitive	variant	TX				RX			
		protected		unprotected		protected		unprotected	
		t-value	energy(mJ)	t-value	energy(mJ)	t-value	energy(mJ)	t-value	energy(mJ)
Syndrome extractor	(31,29,1)RS	0.06	0.016	92.69	0.021	0.73	0.023	2039.85	0.029
Syndrome extractor	(31,27,2)RS	1.64	0.029	2220.64	0.037	1.67	0.043	352.48	0.053
Syndrome extractor	(63,49,7)RS	0.99	0.136	4927.79	0.198	0.80	0.237	438.42	0.308
Pinskech extractor	(255,50,2)	1.01	0.089	242.24	0.046	0.72	0.168	733.98	0.127
Pinskech extractor	(255,20,2)	0.77	0.052	468.67	0.044	0.91	0.208	428.64	0.201
Pinskech extractor	(255,10,2)	0.20	0.066	298.66	0.067	6.63	0.377	236.67	0.380

We empirically verified the effects of implementing the above countermeasures, including the effect on timing leakage and energy consumption (communication overhead included). We used **dudect** [144] to evaluate the timing leakage of the TX and RX implementations running on the MSP430FR5969. The results are given in Table 7.1. The timing leakage of a program is evaluated by Welch’s t-test in **dudect**. For each TX and RX implementation, the t-value in Table 7.1 was computed using 10,000 timing measurements. For a t-value ≤ 10 , **dudect** regards the timing leakage as insignificant given the number of timing measurements.

It is evident that the baseline implementations exhibit strong timing leakage, while the protected variants significantly reduce the leakage below the constant-time threshold of the t-test in **dudect**. Besides, the energy consumption is not significantly increased for the protected variants. In fact, in some cases the energy consumption even decreases because of the use of Barrett reduction, which replaces the costly modulo operation otherwise implemented through division.

7.3 Results

We implemented 22 variants of fuzzy primitives in total for the Hamming metric and three for the set metric. For each variant, we measured its computational energy consumption and estimated the communication cost at both TX and RX sides, and give the input size (length of sc in bits) required to achieve 128-bit security. Table 7.3 shows the main building blocks used by each fuzzy primitive.

In the following, we focus on the evaluation and comparison of selected variants. We include full, detailed results for all variants in Table 7.2. As mentioned, we base our estimation of communication costs on the values of $0.72 \mu\text{J}$ per bit for TX and $0.81 \mu\text{J}$ for RX [141], but also provide the number of exchanged payload bits for use with other models to estimate communication energy.

7.3.1 Hamming Metric Constructions

The minimum number of input bits for sc is given as $n \cdot f \cdot r$, where n is the codeword length of the chosen ECC, f is the number of bits constituting a symbol, and r is the number of repetitions (*i.e.*, how many iterations of the primitive are required for 128-bit security). Depending on the specific variant, between 150 and 420 sc bits are required (see Table 7.2).

Computation Costs. Figure 7.1 shows the energy cost of the Hamming metric fuzzy primitives. For each mismatch threshold, we show four variants of each fuzzy primitive and note that they are adequate to indicate the overall trend.

At TX side, we observe that the fuzzy commitment consumes the least energy. The cost of the code-offset extractor is generally slightly higher than the fuzzy commitment under different mismatch rates. This result is in line with our expectations, because the code-offset extractor can be seen as a fuzzy commitment with additional invocation of a strong extractor. The syndrome fuzzy extractor involves the most energy-intensive computations. This is because syndrome generation for BCH and RS codes is an expensive operation that involves repeatedly evaluating $p(x)$ given x , which requires a number of iterative operations.

Table 7.2 shows the detailed measurement results for all considered fuzzy primitive instantiations in both the Hamming and set metric. We include the following characteristics of each variant: error tolerance, maximum average bit error rate, required number of iterations to achieve 128-bit security, computational energy cost at TX/RX (excluding communication cost), communication overhead (in bits transmitted/received), and required number of bits for sc . * indicates implementation on MSP430FR5994 due to memory requirements.

TABLE 7.2: Evaluation of Hamming and set metric methods

Error tolerance	Max. average bit error rate	Variant	# iterations	sc (bit)	Fuzzy commitment			Code-offset construction			Syndrome construction		
					TX (mJ)	RX (mJ)	Comm (bit)	TX (mJ)	RX (mJ)	Comm (bit)	TX (mJ)	RX (mJ)	Comm (bit)
2%	0.0016%	(50, 44, 1) BCH	3	150	0.022	0.079	150	0.030	0.091	150	0.052	0.084	36
2%	0.0146%	(100, 86, 2) BCH	2	200	0.037	0.201	200	0.045	0.215	200	0.127	0.207	56
2%	0.0854%	(200, 168, 4) BCH	1	200	0.082	0.389	200	0.087	0.404	200	0.244	0.394	64
3.23%	0.0037%	(31, 29, 1) RS	1	155	0.011	0.017	155	0.016	0.021	155	0.014	0.021	10
5%	0.0024%	(20, 15, 1) BCH	9	180	0.029	0.107	180	0.047	0.127	180	0.070	0.120	90
5%	0.0273%	(40, 28, 2) BCH	5	200	0.034	0.225	200	0.046	0.239	200	0.128	0.235	120
5%	0.0854%	(60, 42, 3) BCH	4	240	0.047	0.344	240	0.059	0.357	240	0.221	0.354	144
5%	0.1728%	(80, 52, 4) BCH	3	240	0.076	0.526	240	0.085	0.545	240	0.294	0.533	168
5%	0.2842%	(100, 65, 5) BCH	2	200	0.080	0.505	200	0.091	0.519	200	0.306	0.505	140
5%	0.9005%	(220, 136, 11) BCH	1	220	0.176	0.925*	220	0.179	0.976*	220	0.725	1.096*	176
6.67%	0.0038%	(15, 13, 1) RS	3	180	0.020	0.027	180	0.029	0.041	180	0.026	0.039	24
6.45%	0.0662%	(31, 27, 2) RS	1	155	0.019	0.034	155	0.023	0.040	155	0.023	0.037	20
10%	0.0031%	(10, 6, 1) BCH	22	220	0.060	0.141	220	0.099	0.180	220	0.103	0.180	176
10%	0.0407%	(20, 10, 2) BCH	13	260	0.046	0.307	260	0.073	0.331	260	0.175	0.331	260
10%	0.1429%	(30, 15, 3) BCH	9	270	0.044	0.399	270	0.065	0.420	270	0.259	0.428	270
10%	0.2904%	(40, 16, 4) BCH	8	320	0.067	0.732	320	0.085	0.745	320	0.394	0.735	384
10%	0.4822%	(50, 23, 5) BCH	6	300	0.074	0.786	300	0.091	0.797	300	0.452	0.785	360
10%	0.6855%	(60, 27, 6) BCH	5	300	0.077	0.878	300	0.092	0.896	300	0.548	0.885	360
10%	1.2713%	(90, 34, 9) BCH	4	360	0.131	1.713	360	0.145	1.730	360	0.965	1.724	504
10%	1.8153%	(120, 43, 12) BCH	3	360	0.156	2.102	360	0.168	2.110	360	1.276	2.085	504
10%	3.0090%	(210, 70, 21) BCH	2	420	0.288	3.340*	420	0.306	3.480*	420	2.637	4.025*	672
11.11%	2.4626%	(63, 49, 7) RS	1	378	0.103	0.247	378	0.107	0.252	378	0.138	0.240	84

Error tolerance	Variant	Fuzzy vault					Improved Juels-Sudan construction					Pinskech construction				
		# iterations	TX (mJ)	RX (mJ)	sc (bit)	Comm (bit)	# iterations	TX (mJ)	RX (mJ)	sc (bit)	Comm (bit)	# iterations	TX (mJ)	RX (mJ)	sc (bit)	Comm (bit)
2%	(255, 50, 2)	1	2.534	7.036*	400	4080	1	0.391	18.957*	400	16	1	0.034	0.114	400	16
5%	(255, 20, 2)	1	1.354	0.640	160	4080	2	0.132	2.980	320	32	2	0.021	0.175	320	32
10%	(255, 10, 2)	2	1.906	0.238	160	8160	4	0.073	2.320	320	64	4	0.021	0.328	320	64

TABLE 7.3: Main building blocks of fuzzy primitives for TX and RX.

TX side	RNG	ECC encoding	XOR	Hash	Syndrome gen.	Find poly. coeffs	Gen. points on $p()$	Gen. chaff points	Shuffle points
Fuzzy commitment	●	●	●						
Fuzzy vault	●						●	●	●
Code-offset extractor	●	●	●	●					
Syndrome extractor				●	●				
Improved JS extractor				●		●			
Pinsketch extractor				●	●				
RX side	ECC decoding	XOR	Reconstruct $p()$	Syndrome gen.	Hash	Gen. points on $p()$	Filter points	Find root on $p()$	
Fuzzy commitment	●	●							
Fuzzy vault			●				●		
Code-offset extractor	●	●			●				
Syndrome extractor	●	●		●	●				
Improved JS extractor			●		●	●		●	
Pinsketch extractor	●	●		●	●				

At RX side, we note that the energy consumption of the fuzzy commitment is also the smallest. The difference in computational energy consumption between the code-offset and syndrome extractors is often small because both extractors share several building blocks. Note that, even if the syndrome extractor has an extra “syndrome generation” block compared to the code-offset extractor (see Table 7.3), the actual execution of these constructions is equivalent.

Regarding ECC choice, the RS code performs substantially better than BCH for mismatch rates below 5%. However, for 10% mismatch, the chosen RS code is worse than the best BCH variant. However, note that e.g., the $(63, 49, 7)$ RS instance can accommodate up to 343-bit distribution per iteration for the fuzzy commitment, while we only require 128 bits, which is not optimal if only considering computational energy consumption.

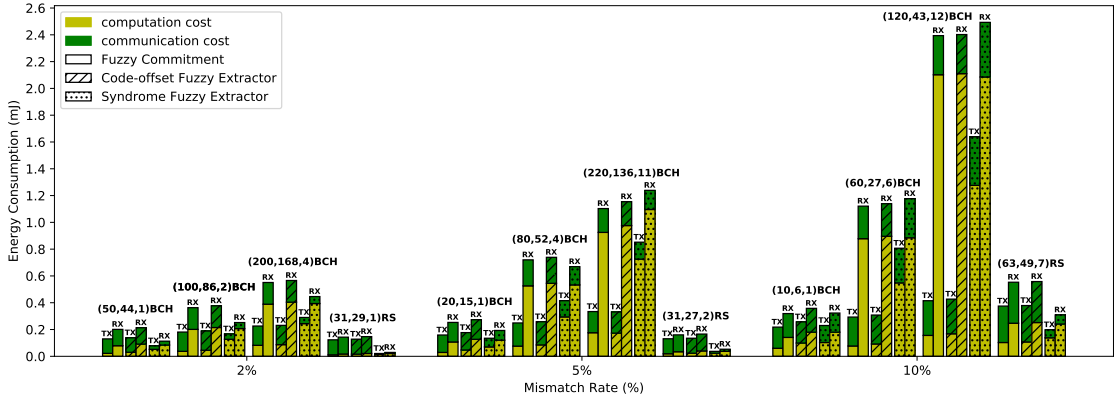


FIGURE 7.1: Energy consumption of Hamming metric primitives.

Combined Computation and Communication Cost. When we also take the estimated communication costs into account, the syndrome fuzzy extractor outperforms the other two variants most of the time. For both fuzzy commitment and code-offset construction, the communication overhead is determined by the codeword length n of the chosen ECC and the number of required repetitions. For example, consider the $(50, 44, 1)$ BCH variant in Table 7.2. In this case, the communication overhead is $50 \cdot 3$

bits, because the variant needs to be executed three times to establish a 128-bit key. In contrast, the communication overhead for the syndrome extractor depends on the syndrome length and the number of repetitions. The length of the syndrome is $2 \cdot t \cdot m$ for BCH and RS codes, where m comes from $GF(2^m)$ underlying the ECC [107]. An obvious advantage is that the syndrome is always shorter than the codeword. Considering the previous example, TX would only need to transmit $12 \cdot 3$ bits for the syndrome extractor. Overall, the variants with lowest combined computation and communication cost under each mismatch rate are shown in Table 7.4. The syndrome extractor has variants with the lowest total energy consumption in all conditions.

TABLE 7.4: Fuzzy primitives with lowest total energy cost on the Hamming metric (Syndrome extractor) and set metric (Pinsketech extractor).

mismatch rate	fuzzy primitive	variant	total energy at TX (mJ)	total energy at RX (mJ)
2%	Syndrome extractor	(31,29,1) RS	0.021	0.029
5%	Syndrome extractor	(31,27,2) RS	0.037	0.053
10%	Syndrome extractor	(63,49,7) RS	0.198	0.308
2%	Pinsketech extractor	(255,50,2)	0.046	0.127
5%	Pinsketech extractor	(255,20,2)	0.044	0.201
10%	Pinsketech extractor	(255,10,2)	0.067	0.380

7.3.2 Set Metric Constructions

The required number of derived bits for set metric methods is $s \cdot f \cdot r$, where s is the set size, f is the number of bits constituting a set element, and r is the number of repetitions. Set metric constructions require input sizes from 160 to 400 bits (see Table 7.2), which is similar to the Hamming metric variants.

Computation Costs. Figure 7.2 shows the energy consumption of all considered set metric methods (see Table 7.2 for the underlying data). At TX, we observe that the fuzzy vault consumes substantially more energy than the other two methods. This is likely because TX of the fuzzy vault has to generate and shuffle a large amount of points (mostly chaff points). The TX energy cost of the Pinsketech fuzzy extractor is slightly below improved Juels–Sudan. According to Table 7.3, the difference in energy consumption is due to the difference between syndrome generation (note that this is not the same as the standard syndrome calculation of BCH and RS code) and the polynomial coefficient finding.

On the RX side, we find that the Pinsketech construction has the lowest energy consumption for mismatch rates below 5%, and has slightly higher consumption than fuzzy vault for 10% mismatch. The improved Juels–Sudan fuzzy extractor is always the most expensive construction under all mismatch rates, likely due to the required polynomial root finding process. We further observe that the energy consumption of this method

and the fuzzy vault decrease significantly when the mismatch rate threshold increases. This is expected because both methods rely on the same complex polynomial reconstruction for mismatch correction. This involves operations on $s \times s$ matrices (s is the set size). Hence, polynomial reconstruction is efficient for small sets (*i.e.*, under higher mismatch threshold), but as the set size increases, the computational complexity increases quadratically.

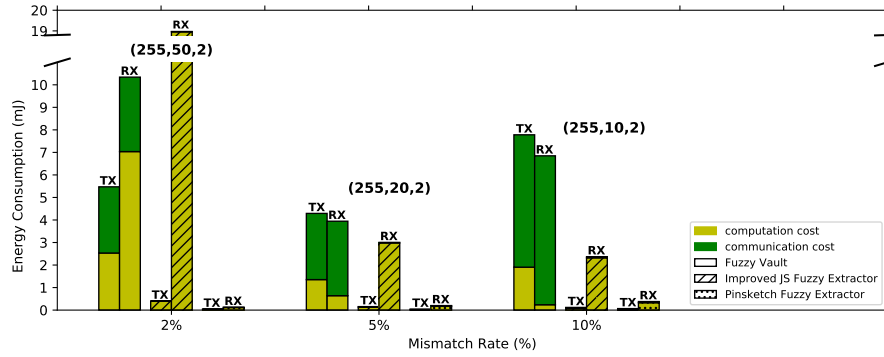


FIGURE 7.2: Energy consumption of set metric primitives.

Combined Computation and Communication Cost. The number of transmitted bits for the fuzzy vault is $np \cdot lp \cdot r$, where np is the total number of points (valid and chaff points), lp is the length of each point, and r is the repetition count. For our universe size of 255, $np = 255$ and $lp = 16$ bits (each coordinate is one byte). In contrast, the transmission size for improved Juels-Sudan and Pinsketech fuzzy extractors is $t \cdot f \cdot r$, where t is the maximum tolerable set difference between sets and f is the number of bits constituting a set element (in our constructions $f = 8$). Hence, the communication cost of the fuzzy vault is much higher compared to improved Juels-Sudan and Pinsketech. The overall best set metric variants for each mismatch threshold are shown in Table 7.4. The Pinsketech fuzzy extractor performs best in terms of combined computation and communication cost in all cases. The improved Juels-Sudan has extremely high computation cost in RX, while the fuzzy vault incurs substantial communication overhead. Considering that secure implementation of fuzzy vault requires operations over $GF(2^{16})$ and transmits more points, it is likely that costs would further grow in practice.

7.4 Discussion and Conclusion

We observed certain common tendencies for all fuzzy primitives: for each variant, the computation energy consumption for RX is generally higher than for TX. Conversely, the communication cost is roughly the same for TX and RX. This observation is relevant when assigning TX/RX roles in more complex protocols; *e.g.*, a low-power IMD can act as TX if the goal is to minimize energy consumption. In addition, the energy

consumption of variants with larger parameter (e.g., larger BCH code) shows an increase both for TX and RX, even though the number of required repetitions decreases.

We compared the energy cost of our best-performing variants with Curve25519, one of the most efficient elliptic curve-based key exchange schemes for embedded systems. As reported in [145], one full execution of Curve25519 on MSP430FR5969 costs about 0.012 mJ (0.404 mJ if communication energy is estimated as in this thesis). Thus, our methods are comparable in terms of total energy consumption.

In this chapter, we systematically and fairly evaluate the performance of fuzzy cryptographic primitives under controlled conditions on a resource-constrained microcontroller. We show how different fuzzy primitives can be securely applied to exchange a key from a shared secret with noise, and provide implementations of each of these primitives in multiple variants. To our knowledge, we are the first to compare the computation and communication energy consumption of different fuzzy primitives for a variety of parameter choices. Among all considered fuzzy primitives, we find that Syndrome and Pinsketch fuzzy extractors overall offer the lowest energy consumption in Hamming and set metric spaces.

This indicates that fuzzy commitment and fuzzy vault commonly used in previous work [138, 146–149] are not optimal on constrained devices. Instead, Syndrome/Pinsketch fuzzy extractors may be preferable, with the added advantage that they neither require random number generation, which can be costly on embedded systems, nor uniformly randomly distributed inputs. These constructions consume between 0.021 mJ and 0.198 mJ for TX and between 0.029 mJ and 0.380 mJ for RX, including computational and communication energy. This demonstrates that using fuzzy primitives in a pairing protocol is feasible for an IMD. Our work in this chapter serves as a reference when applying fuzzy primitives for body-area networks and medical devices, and for other use cases such as biometrics or Physical Unclonable Functions (PUFs).

Chapter 8

Conclusions and Future Directions

In this chapter, we reflect on the research findings presented in this thesis and address the research questions outlined in Chapter 1. We also discuss several important aspects that are often relevant to multiple chapters in this thesis. These include a discussion of the potential side-channel attacks to our proposed pairing protocol, considerations of the necessary user training process and relevant ethical issues, and an identification of our work’s limitations as well as future research directions for IMD security research community.

8.1 Current IMD Pairing Techniques and Drawbacks

RQ1: What are the existing IMD pairing methods used in academia and industry?

Based on our literature review detailed in Chapter 2, we discovered that many traditional pairing methods designed for mobile or IoT devices (such as those based on symmetric/public keys), as well as other approaches (such as using biometrics), do not adequately meet the specific needs of IMDs due to their inherent limitations (discussed in Section 1.1.1) and unique requirements (outlined in Section 2.1.3). On the other hand, the use of OOB channels for IMD pairing presents certain advantages; however, these techniques also frequently entail significant challenges, in terms of security, usability, and deployability (shown in Section 2.2.2).

One particular issue that has been consistently overlooked in academia is the lack of patient perception during the pairing process. We argue that this is crucial in the

IMD context, given that IMDs are life-critical devices and a security breach can be life-threatening. For instance, if a patient is in a crowded public area, such as a bus or a mall, it may be challenging for them to detect in real-time if malicious adversaries attempt to pair with their IMDs, because the pairing process is designed to be transparent to them. In such scenarios, proximity to the patient, or even brief physical contact with them (a necessary step in many previous IMD pairing techniques [28, 29, 33, 34, 38]), could potentially be overlooked by the patient.

To gain a better understanding of this issue, we conducted visits to a cardiology clinic in a hospital, and studied the pairing process with one of the most widely used pacemakers (or even IMDs) worldwide, as mentioned in Chapter 4. We observed a scenario similar to (or even worse than) the academic cases—the actual pairing process was rather imperceptible and invisible. We argue that this could be a potential threat to many patients who currently use these devices.

The security of existing work commonly relies on two assumptions: (i) The adversary cannot reverse-engineer the pairing protocol and know how to impersonate a legitimate external device to pair with the IMD (security by obscurity). This is particularly true for real medical devices; for example, Medtronic never releases the full details of the pairing process with their pacemaker products [9] and strictly forbids the programmer devices from being rented or sold outside hospital clinics. (ii) The adversary cannot reach close proximity to the patient or touch the patient’s body without the patient noticing. And thus, previous pairing methods are often designed to function only within a close distance, and they claim they cannot work outside this distance (although many of them have been successfully attacked beyond this limitation).

For the first assumption, in the cybersecurity community, it is a common consensus that a protocol’s security should not rely on “security by obscurity”, i.e., we should assume that the adversary has full knowledge of the pairing details. In fact, given the continuous series of attacks on commercial IMDs [13–17], pessimistically speaking, it seems inevitable that security breaches of other IMDs are just a matter of time.

The second assumption is to some extent reasonable, as it strikes a balance between security and availability in emergencies, *i.e.*, although the convenience of proximity-based pairing poses some risks from adversaries, it also offers benefits for doctors who need quick access to the patient’s IMD in emergencies. However, we believe that there is room for improvement: Can we design new pairing techniques for IMDs that enhance patient perception while also being simple to execute and complete?

Another important issue with previous academic works is that they often face deployability challenges, and cannot be deployed in existing IMDs on the market or in consumer devices as external devices. Addressing this issue is another major focus of this thesis.

8.2 Developing “Seamful” IMD Pairing Techniques

RQ2: Can the principle of “seamful” design be used to create new pairing protocols for IMDs?

In this thesis, we propose two techniques to achieve seamful pairing with IMDs. The first approach leverages simple bodily motions (body rotation and tapping) as OOB channels for pairing. The second method utilizes highly perceptible vibration signals for pairing, which outperforms previous vibration-based methods regarding deployability in resource-constrained IMDs and other aspects.

In the context of IMD pairing, unlike most previous research work, we implemented prototypes that used our techniques, and thoroughly and empirically evaluated them with human participants. Overall, we find that the workload required to bootstrap a secure pairing is feasible. We estimate that it only requires the user to perform 3 body rotations, or tap the external device on the body 4 times (using our enhanced prototype, as detailed in Section 5.2.3), or attach the device to the body 4 times. These physical interactions could lead to:

- An **FAR** of 0 and an **FRR** below 2%, indicating that the risk posed by adversaries is low, and legitimate users will likely experience very few failures.
- **Entropy** shared between devices is comparable with traditional 4 to 6 digit PINs that are required by Bluetooth pairing and other security systems.
- **Pairing time** costs about 4 to 10 seconds, which strikes a good balance between security (not ephemeral to be imperceptible) and safety (not too long that can pose concerns of emergency availability).
- High user perception level and high **usability**, as verified in our user studies.
- High **deployability** in current modern IMDs, as well as consumer devices (e.g., smartphones) used as the external device.

As a reminder, the user physical interactions act as only a part of our entire pairing protocol (refer to Section 3.1.1). We have also incorporated other cryptographic components to further enhance security, reliability, and deployability. Specifically, a fuzzy primitive

is employed to eliminate potential noise during the pairing process, and a PAKE algorithm is used to improve security and time efficiency. Under this crypto-framework, an adversary in our adversary model is forced to wait for a legitimate party (e.g., the patient or doctor) to perform the motions. At the same time, they need to guess the motions and hope they can succeed in one attempt. The entropy of the performed motions can reduce the adversary’s success probability to lower than 0.01%¹. Note that, if required, higher entropy can be simply achieved by performing more motions.

TABLE 8.1: Comparison with previous work, in terms of accuracy (FAR, FRR) and time efficiency (throughput).

Technique	(FAR, FRR)	Throughput (bps)
Shake well before use [65]	(0, 10 - 12 %)	7
T2Pair [68]	(0, 3 - 9 %)	10.3 - 13.2
Checksum [63]	(5.5 %, 10 %)	3.6
Walkie-Talkie [66]	(-, 0)	12.8
Vibrate-to-unlock [41]	(-, 0)	1.75
H2B [33]	(0, 4.4 %)	3.2
Securing wireless neurostimulators [16]	(-, 0)	85.3
Our rotation prototypes	(0, 1.4 - 1.9 %)	1.32 - 2
Our tap prototypes	(0, 0.1 - 2 %)	7.31 - 8.01
Our vibration prototype	(0, 0.6 %)	3.2

In Table 8.1, we further compare our proposed techniques with prior work, including motion-based pairing and IMD pairing techniques (both elaborated in Chapter 2). Our work outperforms previous studies [33, 63, 65, 68] in terms of accuracy (the others do not provide FAR values and only focus on achieving low FRR). Moreover, the time efficiency of our work is comparable with previous work, except for [16] that uses an electrical current for secret transmission. As mentioned above, for our IMD pairing techniques, the relatively lower data throughput is not necessarily a disadvantage, as it gives patients more time to detect and react to malicious pairing attempts.

Notably, there is always a trade-off in pairing between being seamless or seamful. A seamful design enhances patient perception and sense of control, but it inevitably introduces drawbacks such as increased physical and mental burden on the user. We fully acknowledge this trade-off, and thus conducted several user studies to assess the real impact of these drawbacks. Fortunately, our results show that these negative impacts are limited and might be tolerated by people in practice. Nonetheless, we emphasize that further studies involving a broader demographic are necessary.

Furthermore, we note that the aforementioned user workloads—three rotations, or four taps, or four attachments—are the least amount of human input required to achieve an acceptable level of accuracy and security, which serve as a baseline. Intuitively,

¹In our adversary model (see Section 3.1.3), we assume the adversary cannot observe the motions.

we believe these efforts are perceptible enough for patients; however, in practice, more motions may be needed, especially for elderly patients who may require more time to react to malicious behavior. We leave investigations of the “optimum” amount of motions needed for pairing as future work.

8.3 Deployment in Real-World Healthcare Systems

RQ3: How can our proposed protocols be implemented in such a way so that they can run on existing IMDs?

Our proposed pairing techniques are intended to redesign the pairing process of IMDs; all other aspects of IMD usage and lifecycle largely remain unchanged by our techniques. Thus, we do not expect any interference with other established procedures, like physiological sensing, medical examination, and telemetry functions.

In terms of hardware, our proposed methods only require an inertial sensor, a component already present in the latest generation IMDs, such as those produced by Medtronic [57–59]. Further investigation is warranted to better anticipate how our methods would fare in real devices. In particular, we have not yet evaluated the reliability across sensors of varying quality. For instance, a lower-quality accelerometer with an unstable sampling rate could lead to higher mismatches in the timing of taps. Nevertheless, such issues can be mitigated by incorporating additional motions to ensure accurate pairing.

As for software, the algorithms employed in our prototypes, along with certain cryptographic algorithms like PAKE [68] and fuzzy cryptographic primitives [19, 68], have been validated to efficiently function on 32-bit ARM Cortex-M microcontrollers (or even less advanced hardware, see our work in Chapter 7), which closely resemble the capabilities of modern IMDs [8]. Consequently, the software for our techniques can potentially be deployed into existing IMD models (including the ones that are already implanted in patients) through over-the-air firmware updates [150], but certainly there would be a need for extensive software testing of the implementation before using our approaches in life. We will make all of our developed software available as open source to facilitate this process.

We acknowledge that our methods might need adjustments to better suit individual user preferences and abilities. For our rotation-based pairings, patients with visual impairments could benefit from replacing the visual instructions with auditory ones, as suggested by the pacemaker patient during our co-design workshop. For tap-based approaches, a potentially more comfortable alternative for patients could be attaching the external device to the body and tapping it with fingers, instead of holding it in hand

and tapping against the body. For the vibration-based techniques, further testing of the vibration amplitude and frequency might help determine if they influence the comfort level of the user experience. These potential adaptations warrant further exploration in future studies.

Although our work focuses on pacemakers as a concrete type of IMD, we argue that the techniques can be easily transferred to other IMDs in human bodies (or even wearables), as long as they are equipped with inertial sensors, such as modern insulin pumps [10]. This is because our methods leverage inertial sensors for orientation and vibration measurements, which are independent of the location in the body where they are implanted. For example, taps can also be performed in the waist area for insulin pumps.

Furthermore, we assume the external device is a smartphone in our studies. Note that smartphones cater to general use cases; thus, if the pairing method works with a smartphone, it should also be compatible with proprietary devices developed by medical organizations. Specifically, our vibration-based pairing technique relies solely on vibrations at a constant frequency at the external device side, which can be easily implemented on smartphones and tablets equipped with either an ERM type vibration motor (as used in our prototype) or LRA motors.

We acknowledge that while this thesis emphasizes the practical applicability of the proposed pairing techniques for real-world deployment, most of the work has been conducted within a controlled laboratory environment. In practice, several challenges may arise when deploying these techniques in next-generation IMDs.

For instance, although we have demonstrated that our pairing protocol can run on a 32-bit microcontroller simulating the properties of modern IMDs, we have not yet assessed the actual resource demands of conventional medical tasks (e.g., physiological sensing and wireless communication), which will already consume a significant portion of the system's resources. Another concern is the long-term sustainability of our pairing techniques. For example, while they perform well in controlled environments, scaling them to accommodate a larger number of patients and devices may introduce unforeseen complexities and glitches—issues that must be avoided, especially in this critical context.

To address these challenges, we argue for future work to extend beyond laboratory studies and pursue active collaborations with medical facilities and IMD manufacturers (which was largely hindered by COVID-19 during this thesis). This would involve a more comprehensive assessment of IMD hardware capabilities, optimization of the pairing algorithm for specific hardware architectures, and extensive testing with real-world patient groups. Further details on these aspects are provided in Section 8.6.

8.4 Security Against Side-Channel Attacks

In the previous chapters, we assumed that an adversary in our pairing protocol (Section 3.1.1) is limited to guessing the STK only, a capability largely constrained by the randomness and entropy inherent in user’s performed motions (*i.e.*, *sc* in Figure 3.1). However, due to the use of human physical motions as secrets, our proposed OOB channels inevitably emit a degree of visual and/or auditory (for motion generated noises) side-channel information. Hence, our protocol is in principle vulnerable to advanced eavesdropping attacks based on camera (using computer-vision techniques) or microphone. As a result, the entropy of the motions may be compromised to some extent.

Certainly, this vulnerability is not exclusive to our work; it is a common issue in previous approaches on motion-based pairing with IoT devices [62, 63, 65, 68, 69, 73], and OOB pairing with IMDs, such as those using audio [22], vibration [38, 41], and physiological signals [32, 33, 46].

Various defense strategies have been proposed for these works [38, 43, 55, 61, 68, 151], and such strategies can also be adapted to our protocol. For example, attacks using microphones can be significantly attenuated by adding auditory masking signals to the external device using an integrated speaker [43, 55]. Camera-based attacks can be mitigated by conducting the pairing in private settings, like the clinic or patient’s home [68, 73]. Additionally, visual obfuscation techniques, *e.g.*, those using infrared light [61], could be added into the external device to impair camera visibility of the user.

We emphasize that for our protocol, in the worst case, adversaries capable of launching advanced side-channel attacks can compromise the security *only while* the legitimate parties are present and conducting the pairing. In contrast, a potentially greater threat is the possibility of an adversary silently injecting a key into the patient’s IMD and pairing with it in daily life, a common vulnerability in previous OOB-based IMD pairing methods (see Section 2.2 and Chapter 4). Our use of human physical interactions inherently provides a robust defense against injection attacks. This way, patients in public areas can feel more secure, as the risk of an adversary nearby attempting to pair with their IMD is fundamentally restricted.

8.5 User Training and Ethical Considerations

User Training. Our pairing process involves users executing several motions, thus requiring some training to ensure that they can perform the motions correctly. This

process can conveniently take place in a clinic, such as after IMD implantation for patients, or during dedicated sessions for caregivers and medical professionals. Note that nowadays, training sessions for various medical devices, including IMDs, are common practice in hospitals; in fact, hospitals often have staff who are almost exclusively dedicated to educating patients to use their devices correctly [152, 153]. The creation of accessible, user-friendly educational resources and their validation with people from diverse demographics are therefore of utmost importance.

Nonetheless, our findings imply that the training required for our techniques is rather minimal. Recall that for each pairing technique (either in Chapter 5 or Chapter 6), all participants, including two elderly individuals with limited technological proficiency from our co-design workshop, were ready to proceed to the formal data collection phase after less than one minute of training.

Specifically, we can gain valuable insights from our Study II in Chapter 5: our developed smartphone apps have shown promise to be effective training tools, as evidenced by the participants' satisfactory performance after training and their overall positive experiences. Specifically, participants found that our rotation apps provide intuitive, real-time animations as well as clear visual and tactile feedback, helping them to become acquainted with the pairing technique. For tap motions, as elaborated in Section 5.3.5, the enhanced smartphone app has emerged as a valuable training tool. It is important to note that our study used prototype-level apps, which still hold significant potential for refinement to be dedicated training programmes. Similarly, such training apps can also be developed for our vibration-based pairing technique in Chapter 6.

Ethical Implications. Both ethicists [154] and the HCI community [155] have indicated healthcare as a high-stakes domain when integrating emerging technologies. Particularly, IMD users are often elderly, a group inherently vulnerable associated with their reduced technological literacy and heightened health concerns [156]. Thorough ethical considerations and development of mitigation strategies will ease the integration of our proposed pairing techniques.

User consent demands rigorous attention. To safeguard patients' interests, it is essential to develop comprehensive informed consent processes (as what HCI community has been putting effort to [157–160]). Patients should be fully informed of the role of the required physical interactions in enhancing safety and security. Additionally, patients should have the autonomy to choose whether to complete the pairing themselves, and/or allow doctors to perform the required motions on their behalf. Such robust consent mechanisms can empower vulnerable patients, build their trust in IMDs, and pave the way for more empathetic and secure advancements in healthcare technology.

Healthcare professionals and caregivers play an important role in IMD pairing, as they could guide and monitor the patient, and even carry out the pairing on the patient's behalf. Proper support and education [161] for these people are vital to ensure they are informed and well-prepared for the pairing process. Furthermore, it is crucial for future research to validate whether our proposed techniques overly complicate or extend the duties [162] of healthcare professionals and caregivers. Investigating the potential emotional and psychological effects on them is necessary to ensure the pairing technology does not impose extra stress or burden.

Patient privacy is paramount for IMD pairing [156]. One major concern is the inadvertent revelation of a user's IMD status when observed performing the required operations for pairing. To mitigate this, pairing can be conducted in private places, such as a clinic or the patient's home. Moreover, the motions executed during pairing might reveal the user's personal habits. However, these motions function only as an ephemeral secret and can be permanently erased once the pairing is complete. Thus, any information disclosed during the pairing process poses a limited risk to the user's privacy. We advocate for deeper investigations of other potential privacy issues and their corresponding solutions in future work.

In terms of data security, our pairing techniques are designed to complement existing security measures in conventional IMDs [9]. Our methods focus on the secure pairing process and do not interfere with standard data security practices like encryption and secure storage. This ensures that the integrity and confidentiality of patient data are maintained, aligning with the security policies typically used in current medical devices [156].

Health Impacts. Our proposed pairing techniques involve bodily motions and vibrations performed on the patient's body, which may raise concerns about potential long-term health impacts. During our co-design workshop (see Section 5.3.7), the IMD experts confirmed that the required body rotations and taps pose minimal health risks to patients. Regarding vibrations, existing research indicates that only prolonged and excessive exposure (e.g., in certain occupational environments) is linked to adverse mental and physical health effects [133, 134]. In contrast, our method uses brief vibrational interactions lasting less than five seconds. Additionally, pairing operations would likely occur infrequently—approximately once a month during clinic visits—further reducing the likelihood of any negative health outcomes.

Nevertheless, we recognize the critical importance of thoroughly understanding any potential health implications of our approach, especially in the delicate context of patient care. To address this, future collaborations with a broader range of medical experts

(such as physicians and IMD industry representatives) will be essential for comprehensive evaluations, ensuring that any health risks are rigorously assessed and mitigated.

8.6 Limitations and Future Directions

Our work presented in this thesis has a number of limitations. This section describes them and points out suggestions for potential future work.

One is the participants' demographic, as we did not specifically recruit IMD patients, mainly due to ethical constraints of the institution where the study was conducted². For example, patients with cardiac implants are often elderly, usually around 63 to 75 years old according to [48]. While we did run a co-design workshop with an IMD patient and an expert from this age group for our Study II, the majority of our participants did not fall within this demographic. This might potentially introduce biases in the design and evaluation of our proposed techniques. For instance, other elderly individuals may perform the required motions differently, e.g., rotations may be slower or more error-prone. Thus, future studies should validate our techniques in more diverse populations, including more elderly IMD patients.

Similarly, while our initial investigations in a clinical setting provided valuable insights (see Chapter 4), subsequent studies were confined to a controlled laboratory environment. Another aspect of future work is to iteratively evaluate and optimize our techniques with medical domain experts, like doctors, emergency workers, and medical company representatives. Their participation is essential to identify and rectify potential design biases, and to ensure our methods are well-suited to the IMD context.

Another limitation concerns the adversary model. As mentioned earlier, our work involves user behavior associated with secrets, making it potentially susceptible to advanced side-channel attacks using cameras or microphones to capture the legitimate user's behavior in real-time. Although these types of attacks are not explored in this thesis, their impact on our specific work and the effectiveness of previous mitigation solutions warrant future investigation, particularly in light of the increasing capabilities of artificial intelligence.

Additionally, although we designed our prototype in line with previous work in the IMD security community [16, 22, 38], there is room for enhancement particularly in its size and weight. Future research should develop more skin-conformable and miniaturized prototypes [163, 164]. Nevertheless, we believe that the current imperfections of our prototype do not compromise the validity of our results.

²This is particularly strict during the COVID-19 pandemic.

Moreover, during our literature review, we observed that while many studies have examined the vulnerabilities of existing IMD products in the market, there is a scarcity of work that summarizes and compares the specific security solutions employed by these devices in detail. We believe that exploring this area could be a valuable direction for future research, as it would provide a clearer understanding of the current security landscape in IMDs and potentially identify areas for improvement.

Furthermore, we emphasize that although substantial work has proposed techniques for pairing with IMDs, the primary focus was on security aspects. The perspectives of key stakeholders in this context—namely patients, medical personnel, and medical providers—have largely been overlooked. [113] is the first study that delved into this area. The authors interviewed 13 patients with IMDs and explored their views and values regarding security of their devices. This information was then used to inform the design of pairing mechanisms for IMDs. Another work conducted workshops with 24 medical providers, including cardiologists, physicians and medical device manufacturer representatives [165]. The aim was to gather their views on previously proposed security solutions for cardiac implants within academia. Both studies have highlighted that while certain techniques may be technically feasible and sound from a technician or security expert’s standpoint, they may not be desirably received or considered practical by patients and medical staff. We argue for more future work like these ones.

Finally, it is worth pointing out that while the concept of “seamful” pairing discussed in this thesis was developed for IMDs, it can be extended to a wide range of other applications. For instance, consider a user who owns a Bluetooth headphone and multiple terminal devices such as a smartphone, laptop, and PC. Typically, when the headphone is powered on, it automatically connects to the last device it was paired with. However, this default behavior may not always match the user’s preferences, necessitating manual reconnection to the desired device—a process that can be quite inconvenient. By incorporating seamful interaction for device pairing, the user could simply tap the headphone several times against the target terminal to establish a new connection.

Another potential use case involves scenarios where users prefer to pair with a device only when they are physically co-located with it. In these cases, the need for physical contact ensures that only devices in close proximity can be paired and interacted with. This adds a layer of intentionality and security to the pairing process. Overall, we believe that the core concepts proposed in this thesis could provide valuable insights for designing the next generation of Human-Computer Interaction systems.

8.7 Conclusion

In this thesis, we investigate the existing techniques for pairing with an IMD, encompassing approaches from both academia and those employed by real-world medical devices. We identify concerns regarding the security and deployability of current techniques. Specifically, we highlight a significant issue that has been consistently overlooked in existing work: the pairing process is often designed to be “seamless”. Although this provides convenience to legitimate users (such as the patient and doctor), it also makes malicious pairing attempts towards IMDs difficult to detect, potentially leading to severe consequences in this context.

To address the limitations of previous work, in this thesis, we introduce a new user-driven pairing protocol between an IMD and an external device. This protocol incorporates novel pairing designs that involve a degree of physical interaction with the patient’s body. For instance, the pairing can be completed by the patient (or a doctor) tapping the external device on the patient’s body for a few times. We thoroughly validate the feasibility and usability of these techniques with human participants, and found that secure pairing can be achieved conveniently. This approach maintains the benefits of previous pairing techniques such as high usability, while significantly increasing the patient’s perception level and, consequently, enhancing security.

The thesis then explores the deployability of our proposed pairing protocol. Specifically, we verify that each component of the protocol can operate on embedded systems that mimic the hardware capabilities of modern IMDs. The results further demonstrate that our proposal is promising, and could potentially be integrated into existing IMD products in the market with minor modifications. Certainly, comprehensive software optimization and testing would be necessary for such integration, which is beyond the scope of this thesis. We will provide all involved software and prototype designs as open source to facilitate this process.

Overall, we point out a new direction in this thesis: secure user-driven pairing with IMDs, explaining why it is desired and how it can be achieved. We believe that our work establishes a new reference for future user-friendly design of security solutions for IMDs. We hope that our contributions can offer valuable insights to medical organizations in the development of future IMD products, which will impact the lives of a large group of patients and families.

Bibliography

- [1] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.
- [2] Shoichi Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. In *FSE*, pages 210–225, 2006.
- [3] S.M. Bellovin and M. Merritt. Encrypted key exchange: password-based protocols secure against dictionary attacks. In *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 72–84, 1992.
- [4] Renesas Electronics. Da14580, bluetooth low energy 4.2 soc, rev. 3.5, 2022. <https://www.renesas.com/us/en/document/dst/da14580-datasheet>.
- [5] GlobeNewswire. Global active implantable medical devices market is expected to reach usd 30.42 billion by 2025, 2019. <https://www.globenewswire.com/news-release/2019/08/08/1898922/0/en/Global-Active-Implantable-Medical-Devices-Market-is-Expected-to-Reach-USD-30-42-Billion-by-2025-Fior-Markets.html>.
- [6] Medtronic. Medtronic receives fda approval for its next generation micra leadless pacing systems, 2023. <https://news.medtronic.com/2023-05-01-Medtronic-receives-FDA-approval-for-its-next-generation-Micra-leadless-pacing-systems>.
- [7] Medtronic. Micra: the leadless pacemaker, n.d.. <https://www.medtronic.com/uk-en/c/emea/cardiac-rhythm/micra-leadless-pacemaker.html>.
- [8] Medtronic. Azure pacing system, 2022. <https://europe.medtronic.com/xd-en/healthcare-professionals/products/cardiac-rhythm/pacemakers/azure.html>.
- [9] Medtronic. Bluesync technology within implantable cardiac devices, 2020. <https://europe.medtronic.com/xd-en/healthcare-professionals/therapies-pro>

- [cedures/cardiac-rhythm/cardiac-device-features/bluesync-technology.html](#).
- [10] Medtronic. Insulin pump systems, 2021. <https://www.medtronic.com/us-en/healthcare-professionals/products/diabetes/insulin-pump-systems.html>.
 - [11] Medtronic. Mycarelink mobile app, 2021. <https://global.medtronic.com/xg-en/mobileapps/patient-caregiver/cardiac-monitoring/mycarelink-heart-app.html>.
 - [12] Michele De Santis and Ilaria Cacciotti. Wireless implantable and biodegradable sensors for postsurgery monitoring: Current status and future perspectives. *Nanotechnology*, 31(25):252001, 2020.
 - [13] Luca Reverberi and David Oswald. Breaking (and fixing) a widely used continuous glucose monitoring system. In *Proceedings of the 11th USENIX Conference on Offensive Technologies*, WOOT’17, page 18, USA, 2017. USENIX Association.
 - [14] Eduard Marin, Dave Singelée, Flavio D. Garcia, Tom Chothia, Rik Willems, and Bart Preneel. On the (in)security of the latest generation implantable cardiac defibrillators and how to secure them. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ACSAC ’16, page 226–236, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450347716.
 - [15] Eduard Marin, Dave Singelée, Bohan Yang, Ingrid Verbauwhede, and Bart Preneel. On the feasibility of cryptography for a wireless insulin pump system. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, CODASPY ’16, page 113–120, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450339353.
 - [16] Eduard Marin, Dave Singelée, Bohan Yang, Vladimir Volski, Guy A. E. Vandenbosch, Bart Nuttin, and Bart Preneel. Securing wireless neurostimulators. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY ’18, page 287–298, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356329.
 - [17] Daniel Halperin, Thomas S Heydt-Benjamin, Kevin Fu, Tadayoshi Kohno, and William H Maisel. Security and privacy for implantable medical devices. *IEEE pervasive computing*, 7(1):30–39, 2008.
 - [18] Naked Security. Doctors disabled wireless in dick cheney’s pacemaker to thwart hacking., 2013. <https://nakedsecurity.sophos.com/2013/10/22/doctors-disabled-wireless-in-dick-cheney-s-pacemaker-to-thwart-hacking/>.

- [19] Mo Zhang, Eduard Marin, David Oswald, and Dave Singelée. Fuzzykey: Comparing fuzzy cryptographic primitives on resource-constrained devices. In *Smart Card Research and Advanced Applications*, pages 289–309, 2022. ISBN 978-3-030-97348-3.
- [20] Eduardo Arrais Rocha, Gisele Schineider Cunha, Aline Bezerra Tavares, Antônio Brazil Viana Júnior, Ana Rosa Pinto Quidute, Francisca Tatiana Moreira Pereira, Marcelo de Paula Martins Monteiro, Maria Eduarda Quidute Arrais Rocha, Camila Rabelo Ferreira Gomes, and Carlos Roberto Martins Rodrigues Sobrinho. Syncope in patients with cardiac pacemakers. *Brazilian Journal of Cardiovascular Surgery*, 36:18–24, 2020.
- [21] Wi-Fi Alliance. Wi-fi certified wi-fi direct, personal, portable wi-fi® technology. *White paper*, page 26, 2010.
- [22] 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 *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 129–142, 2008.
- [23] Tim Dierks and Christopher Allen. The tls protocol version 1.0. Technical report, 1999.
- [24] Kasper Bonne Rasmussen, Claude Castelluccia, Thomas S. Heydt-Benjamin, and Srdjan Capkun. Proximity-based access control for implantable medical devices. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, page 410–419, 2009. ISBN 9781605588940.
- [25] Meghan K Hamwey, Emily P Rolan, Alexander C Jensen, and Shawn D White-man. “absence makes the heart grow fonder”: A qualitative examination of sibling relationships during emerging adulthood. *Journal of Social and Personal Relationships*, 36(8):2487–2506, 2019.
- [26] Shyamnath Gollakota, Haitham Hassanieh, Benjamin Ransford, Dina Katabi, and Kevin Fu. They can hear your heartbeats: Non-invasive security for implantable medical devices. In *Proceedings of the ACM SIGCOMM 2011 Conference*, page 2–13, 2011. ISBN 9781450307970.
- [27] Fengyuan Xu, Zhengrui Qin, Chiu C Tan, Baosheng Wang, and Qun Li. Imdguard: Securing implantable medical devices with the external wearable guardian. In *2011 Proceedings IEEE INFOCOM*, pages 1862–1870. IEEE, 2011.

- [28] Marc Roeschlin, Ivan Martinovic, and Kasper Bonne Rasmussen. Device pairing at the touch of an electrode. In *NDSS*, volume 18, pages 18–21, 2018.
- [29] William J. Tomlinson, Stella Banou, Christopher Yu, Michele Nogueira, and Kaushik R. Chowdhury. Secure on-skin biometric signal transmission using galvanic coupling. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 1135–1143, 2019.
- [30] Boston Scientific. Boston scientific electromagnetic (emi) compatibility table for pacemakers, transvenous icds, s-icds and heart failure devices. *Tech Guide*, 2017.
- [31] Kasper Rasmussen, Marc Roeschlin, Ivan Martinovic, and Gene Tsudik. Authentication using pulse-response biometrics. 2014.
- [32] Carmen CY Poon, Yuan-Ting Zhang, and Shu-Di Bao. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communications Magazine*, 44(4):73–81, 2006.
- [33] 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*, IPSN '19, page 265–276, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362849.
- [34] Masoud Rostami, Ari Juels, and Farinaz Koushanfar. Heart-to-Heart (H2H): Authentication for Implanted Medical Devices. In *ACM CCS*, pages 1099–1112, 2013.
- [35] Eduard Marin, Enrique Argones Rúa, Dave Singelée, and Bart Preneel. On the difficulty of using patient’s physiological signals in cryptographic protocols. In *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, SACMAT '19, page 113–122, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367530.
- [36] Lara Ortiz-Martin, Pablo Picazo-Sanchez, Pedro Peris-Lopez, and Juan Tapiador. Heartbeats do not make good pseudo-random number generators: An analysis of the randomness of inter-pulse intervals. *Entropy*, 20(2):94, 2018.
- [37] Robert Mark Seepers, Wenjin Wang, Gerard De Haan, Ioannis Sourdis, and Christos Strydis. Attacks on heartbeat-based security using remote photoplethysmography. *IEEE journal of biomedical and health informatics*, 22(3):714–721, 2017.
- [38] Younghyun Kim, Woo Suk Lee, Vijay Raghunathan, Niraj K. Jha, and Anand Raghunathan. Vibration-based secure side channel for medical devices. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2015.

- [39] Tzipora Halevi and Nitesh Saxena. On pairing constrained wireless devices based on secrecy of auxiliary channels: The case of acoustic eavesdropping. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, page 97–108, 2010. ISBN 9781450302456.
- [40] Inwook Hwang, Jongman Seo, Myongchan Kim, and Seungmoon Choi. Vibrotactile perceived intensity for mobile devices as a function of direction, amplitude, and frequency. *IEEE Transactions on Haptics*, 6(3):352–362, 2013.
- [41] 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)*, pages 181–188, 2011.
- [42] Cybersecurity and Infrastructure Security Agent. Medtronic 2090 carelink programmer vulnerabilities (update c), 2020. <https://www.cisa.gov/uscert/ics/advisories/ICSMA-18-058-01>.
- [43] S Abhishek Anand and Nitesh Saxena. Coresident evil: Noisy vibrational pairing in the face of co-located acoustic eavesdropping. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, page 173–183, 2017. ISBN 9781450350846.
- [44] Jean-Francois Dhem, Francois Koeune, Philippe-Alexandre Leroux, Patrick Mestré, Jean-Jacques Quisquater, and Jean-Louis Willems. A practical implementation of the timing attack. In *Smart Card Research and Applications: Third International Conference, CARDIS'98, Louvain-la-Neuve, Belgium, September 14-16, 1998. Proceedings 3*, pages 167–182. Springer, 2000.
- [45] Michael Rushanan, Aviel D. Rubin, Denis Foo Kune, and Colleen M. Swanson. Sok: Security and privacy in implantable medical devices and body area networks. In *2014 IEEE Symposium on Security and Privacy*, pages 524–539, 2014.
- [46] 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*, page 1099–1112, 2013. ISBN 9781450324779.
- [47] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983. doi: 10.1109/TIT.1983.1056650.
- [48] Wei-Yao Lim, Sandeep Prabhu, and Richard J Schilling. Implantable cardiac electronic devices in the elderly population. *Arrhythmia & Electrophysiology Review*, 8(2):143, 2019.

- [49] Kasper Bonne Rasmussen and Srdjan Capkun. Realization of $\{\text{RF}\}$ distance bounding. In *19th USENIX Security Symposium (USENIX Security 10)*, pages 389–402, 2010.
- [50] S Sedihpour, Srdjan Capkun, Saurabh Ganeriwal, and Mani Srivastava. Implementation of attacks on ultrasonic ranging systems. In *ACM SenSys*, volume 10, pages 1098918–1098977, 2005.
- [51] Tamara Denning, Kevin Fu, and Tadayoshi Kohno. Absence makes the heart grow fonder: New directions for implantable medical device security. In *Proceedings of the 3rd Conference on Hot Topics in Security*, 2008.
- [52] Stuart Schechter. Security that is meant to be skin deep: Using ultraviolet micropigmentation to store Emergency-Access keys for implantable medical devices. In *1st USENIX Workshop on Health Security and Privacy (HealthSec 10)*, Washington, DC, August 2010. USENIX Association.
- [53] Muhammad Ali Siddiqi, Robert HSH Beurskens, Pieter Kruizinga, Chris I De Zeeuw, and Christos Strydis. Securing implantable medical devices using ultrasound waves. *IEEE Access*, 9:80170–80182, 2021.
- [54] Marc Simon Wegmueller, Sonja Huclova, Juerg Froehlich, Michael Oberle, Norbert Felber, Niels Kuster, and Wolfgang Fichtner. Galvanic coupling enabling wireless implant communications. *IEEE Transactions on Instrumentation and Measurement*, 58(8):2618–2625, 2009.
- [55] S Abhishek Anand and Nitesh Saxena. Noisy vibrational pairing of iot devices. *IEEE Transactions on Dependable and Secure Computing*, 16(3):530–545, 2018.
- [56] Eduard Marin, Mustafa A. Mustafa, Dave Singelée, and Bart Preneel. A privacy-preserving remote healthcare system offering end-to-end security. In Nathalie Mitton, Valeria Loscri, and Alexandre Mouradian, editors, *Ad-hoc, Mobile, and Wireless Networks*, pages 237–250, Cham, 2016. Springer International Publishing. ISBN 978-3-319-40509-4.
- [57] Medtronic. Medtronic sensor, n.d.. <https://www.cardiocases.com/en/pacingdefibrillation/specificities/programming-exercise/medtronic/medtronic-sensor>.
- [58] Medtronic. Rate response (rr) feature, 2016. <https://www.medtronicacademy.com/features/rate-response-rr-feature>.
- [59] Venkata K Puppala, Benjamin C Hofeld, Amberly Anger, Sudhi Tyagi, Scott J Strath, Judith Fox, Marcie G Berger, Kwang Woo Ahn, and Michael E Widlansky.

- Pacemaker detected active minutes are superior to pedometer-based step counts in measuring the response to physical activity counseling in sedentary older adults. *BMC geriatrics*, 20:1–11, 2020.
- [60] Achraf Ben Amar, Ammar B Kouki, and Hung Cao. Power approaches for implantable medical devices. *Sensors*, 15(11):28889–28914, 2015.
- [61] HACKADAY. Using ir leds to hide in plain sight, 2020. <https://hackaday.com/2020/02/28/using-ir-leds-to-hide-in-plain-sight/>.
- [62] Tengxiang Zhang, Xin Yi, Ruolin Wang, Yuntao Wang, Chun Yu, Yiqin Lu, and Yuanchun Shi. Tap-to-pair: associating wireless devices with synchronous tapping. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(4):1–21, 2018.
- [63] Imtiaj Ahmed, Yina Ye, Sourav Bhattacharya, N. 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*, page 391–401, 2015. ISBN 9781450335744.
- [64] 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)*, pages 836–852. IEEE, 2018.
- [65] Rene Mayrhofer and Hans Gellersen. Shake well before use: Authentication based on accelerometer data. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive Computing*, pages 144–161, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-72037-9.
- [66] 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 *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*, 2016.
- [67] Girish Revadigar, Chitra Javali, Weitao Xu, Athanasios V Vasilakos, Wen Hu, and Sanjay Jha. Accelerometer and fuzzy vault-based secure group key generation and sharing protocol for smart wearables. *IEEE Transactions on Information Forensics and Security*, 12(10):2467–2482, 2017.
- [68] 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*, CCS '20, page 309–323, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370899.
- [69] 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)*, pages 181–189, 2014.
- [70] Alan Godfrey, AK Bourke, GM Olaighin, P Van De Ven, and J Nelson. Activity classification using a single chest mounted tri-axial accelerometer. *Medical engineering & physics*, 33(9):1127–1135, 2011.
- [71] Matthew Stenerson, Fraser Cameron, Darrell M Wilson, Breanne Harris, Shelby Payne, B Wayne Bequette, and Bruce A Buckingham. The impact of accelerometer and heart rate data on hypoglycemia mitigation in type 1 diabetes. *Journal of diabetes science and technology*, 8(1):64–69, 2014.
- [72] Matthew Stenerson, Fraser Cameron, Shelby R Payne, Sydney L Payne, Trang T Ly, Darrell M Wilson, and Bruce A Buckingham. The impact of accelerometer use in exercise-associated hypoglycemia prevention in type 1 diabetes. *Journal of diabetes science and technology*, 9(1):80–85, 2014.
- [73] 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*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361699.
- [74] Jean-Luc Danger, Youssef El Housni, Adrien Facon, Cheikh Thiecoumba Gueye, Sylvain Guilley, Sylvie Herbel, Ousmane Ndiaye, Edoardo Persichetti, and Alexander Schaub. On the Performance and Security of Multiplication in $GF(2^N)$. *Cryptography*, 2(3):25, 2018.
- [75] Robert M. Seepers, Christos Strydis, Pedro Peris-Lopez, Ioannis Sourdis, and Chris I. De Zeeuw. Peak misdetection in heart-beat-based security: Characterization and tolerance. In *EMBC*, pages 5401–5405, 2014.
- [76] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, CCS '99, page 28–36, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581131488.
- [77] Ari Juels and Madhu Sudan. A Fuzzy Vault Scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006. ISSN 0925-1022.

- [78] Jonathan Katz, Rafail Ostrovsky, and Moti Yung. Forward secrecy in password-only key exchange protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks*, pages 29–44, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. ISBN 978-3-540-36413-9.
- [79] C. Bisdikian. An overview of the bluetooth wireless technology. *IEEE Communications Magazine*, 39(12):86–94, 2001. doi: 10.1109/35.968817.
- [80] Healthline. Everything you need to know about the pacemaker, 2021. <https://www.healthline.com/health/heart-pacemaker#types>.
- [81] Bosch Sensortec. Intelligent 9-axis absolute orientation sensor. *BNO055 datasheet*, November, 2014.
- [82] InvenSense. Mpu-6000 and mpu-6050 product specification revision 3.4, 2013. <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>.
- [83] Wang Wei, Lin Yang, and Qian Zhang. Resonance-based secure pairing for wearables. *IEEE Transactions on Mobile Computing*, 17(11):2607–2618, 2018.
- [84] Anna S. Petronio, Jan-Malte Sinning, Nicolas Van Mieghem, Giulio Zucchelli, Georg Nickenig, Raffi Bekeredjian, Johan Bosmans, Francesco Bedogni, Marian Branny, Karl Stangl, Jan Kovac, Molly Schiltgen, Stacia Kraus, and Peter de Jaegere. Optimal implantation depth and adherence to guidelines on permanent pacing to improve the results of transcatheter aortic valve replacement with the medtronic corevalve system: The corevalve prospective, international, post-market advance-ii study. *JACC: Cardiovascular Interventions*, 8(6):837–846, 2015. ISSN 1936-8798.
- [85] Michael E Schuckers and Michael E Schuckers. Receiver operating characteristic curve and equal error rate. *Computational Methods in Biometric Authentication: Statistical Methods for Performance Evaluation*, pages 155–204, 2010.
- [86] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, M Levenson, M Vangel, D Banks, Nathanael Heckert, James Dray, and S Vo. A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2001-05-15 2001.
- [87] MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.
- [88] Meltem Sonmez, Elaine Barker, John Kelsey, Kerry McKay, Mary Baish, and Mike Boyle. Recommendation for the entropy sources used for random bit generation, 2018-01-10 2018.

- [89] James R Lewis. The system usability scale: past, present, and future. *International Journal of Human-Computer Interaction*, 34(7):577–590, 2018.
- [90] VivaChek Biotech. Vivachek glucose meter systems, n.d. <https://www.vivachek.com/en/products.html>.
- [91] STMicroelectronics. Stm32f0 entry-level arm® cortex®-m0 mcus, 2024. <https://www.st.com/en/microcontrollers-microprocessors/stm32f0-series.html>.
- [92] Adafruit. Introducing the adafruit bluefruit le friend, 2024. <https://learn.adafruit.com/introducing-adafruit-ble-bluetooth-low-energy-friend/introduction>.
- [93] Jay Beale, Angela Orebaugh, and Gilbert Ramirez. *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.
- [94] Sode Pallavi and V Anantha Narayanan. An overview of practical attacks on ble based iot devices and their security. In *2019 5th international conference on advanced computing & communication systems (ICACCS)*, pages 694–698. IEEE, 2019.
- [95] Peter Mell and Tim Grance. Use of the common vulnerabilities and exposures (cve) vulnerability naming scheme. *NIST Special Publication*, 800:51, 2002.
- [96] Rajdeep Bhanot and Rahul Hans. A review and comparative analysis of various encryption algorithms. *International Journal of Security and Its Applications*, 9(4):289–306, 2015.
- [97] Juzheng Liu, Jing Chen, Hanjun Jiang, Wen Jia, Qingliang Lin, and Zhihua Wang. Activity recognition in wearable ecg monitoring aided by accelerometer data. In *2018 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–4. IEEE, 2018.
- [98] Stavros Garzonis, Simon Jones, Tim Jay, and Eamonn O’Neill. Auditory icon and earcon mobile service notifications: intuitiveness, learnability, memorability and preference. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1513–1522, 2009.
- [99] Python Software Foundation. Python secrets library, 2023. <https://docs.python.org/3/library/secrets.html>.
- [100] Imre Lahdelma and Tuomas Eerola. Cultural familiarity and musical expertise impact the pleasantness of consonance/dissonance but not its perceived tension. *Scientific reports*, 10(1):8693, 2020.

- [101] Patrick O'Toole, Donald Glowinski, Ian Pitt, and Maurizio Mancini. When emotions are triggered by single musical notes: Revealing the underlying factors of auditory-emotion associations. In *Companion Publication of the 2021 International Conference on Multimodal Interaction, ICMI '21 Companion*, page 291–298, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384711. doi: 10.1145/3461615.3485419. URL <https://doi.org/10.1145/3461615.3485419>.
- [102] Peter Susic. What is midrange & how mid frequencies impact your music, 2023. <https://headphonesaddict.com/what-is-midrange/>.
- [103] Ben Beaumont-Thomas. The roland tr-808: the drum machine that revolutionised music, 2014. <https://www.theguardian.com/music/2014/mar/06/roland-tr-808-drum-machine-revolutionised-music>.
- [104] Ran Halprin and Moni Naor. Games for extracting randomness. In *Proceedings of the 5th Symposium on Usable Privacy and Security, SOUPS '09*, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605587363.
- [105] Zhirong Lin, Yongsheng Xiong, Houde Dai, and Xuke Xia. An experimental performance evaluation of the orientation accuracy of four nine-axis mems motion sensors. In *2017 5th International Conference on Enterprise Systems (ES)*, pages 185–189, 2017.
- [106] Oracle. Java class securerandom, 2023. <https://docs.oracle.com/javase/8/docs/api/java/security/SecureRandom.html>.
- [107] Shu Lin and Daniel J Costello. *Error control coding*, volume 2. Prentice hall, 2001.
- [108] Lara Ortiz-Martin, Pablo Picazo-Sanchez, and Pedro Peris-Lopez. Are the inter-pulse intervals of an ecg signal a good source of entropy? an in-depth entropy analysis based on nist 800-90b recommendation. *Future Generation Computer Systems*, 105:346–360, 2020.
- [109] Duygu Karaoğlu Altop, Albert Levi, and Volkan Tuzcu. Deriving cryptographic keys from physiological signals. *Pervasive and Mobile Computing*, 39:65–79, 2017.
- [110] Robert M Seepers, Christos Strydis, Ioannis Sourdis, and Chris I De Zeeuw. Enhancing heart-beat-based security for mhealth applications. *IEEE journal of biomedical and health informatics*, 21(1):254–262, 2015.
- [111] Sandeep Pirbhulal, Heye Zhang, Wanqing Wu, Subhas Chandra Mukhopadhyay, and Yuan-Ting Zhang. Heartbeats based biometric random binary sequences generation to secure wireless body sensor networks. *IEEE Transactions on Biomedical Engineering*, 65(12):2751–2759, 2018.

- [112] Sanaz Rahimi Moosavi, Ethiopia Nigussie, Seppo Virtanen, and Jouni Isoaho. Cryptographic key generation using ecg signal. In *2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 1024–1031. IEEE, 2017.
- [113] Tamara Denning, Alan Borning, Batya Friedman, Brian T. Gill, Tadayoshi Kohno, and William H. Maisel. Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 917–926, 2010. ISBN 9781605589299.
- [114] Sarah Inman and David Ribes. "beautiful seams" strategic revelations and concealments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2019.
- [115] Matthew Chalmers and Ian MacColl. Seamful and seamless design in ubiquitous computing. In *Workshop at the crossroads: The interaction of HCI and systems issues in UbiComp*, volume 8, 2003.
- [116] Nitesh Saxena, Md Borhan Uddin, and Jonathan Voris. Treat'em like other devices: user authentication of multiple personal rfid tags. In *SOUPS*, volume 9, pages 1–1, 2009.
- [117] Kyuin Lee, Vijay Raghunathan, Anand Raghunathan, and Younghyun Kim. Syncvibe: Fast and secure device pairing through physical vibration on commodity smartphones. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pages 234–241. IEEE, 2018.
- [118] Robert Xiao, Sven Mayer, and Chris Harrison. Vibrocomm: Using commodity gyroscopes for vibroacoustic data reception. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 1–9, 2020.
- [119] 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)*, pages 265–278, 2015.
- [120] Nirupam Roy and Romit Roy Choudhury. Ripple {II}: Faster communication through physical vibration. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 671–684, 2016.
- [121] Precision Microdrives. Vibration motors – erms and lras, 2021. <https://www.precisionmicrodrives.com/vibration-motors-erms-and-lras>.

- [122] Aftab Khan, Nils Hammerla, Sebastian Mellor, and Thomas Plötz. Optimising sampling rates for accelerometer-based human activity recognition. *Pattern Recognition Letters*, 73:33–40, 2016.
- [123] Gerald Bieber, Thomas Kirste, and Michael Gaede. Low sampling rate for physical activity recognition. In *Proceedings of the 7th international conference on pervasive technologies related to assistive environments*, pages 1–8, 2014.
- [124] Michael Hamman De Vaal, James Neville, Jacques Scherman, Peter Zilla, Micah Litow, and Thomas Franz. The in vivo assessment of mechanical loadings on pectoral pacemaker implants. *Journal of biomechanics*, 43(9):1717–1722, 2010.
- [125] SA Adewusi, Subhash Rakheja, Patrice Marcotte, and J Boutin. Vibration transmissibility characteristics of the human hand–arm system under different postures, hand forces and excitation levels. *Journal of sound and vibration*, 329(14):2953–2971, 2010.
- [126] S Adewusi, M Thomas, VH Vu, and W Li. Modal parameters of the human hand–arm using finite element and operational modal analysis. *Mechanics & Industry*, 15(6):541–549, 2014.
- [127] Li Zhang, Parth H Pathak, Muchen Wu, Yixin Zhao, and Prasant Mohapatra. Accelword: Energy efficient hotword detection through accelerometer. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 301–315, 2015.
- [128] Robert W Doran. The gray code. Technical report, Department of Computer Science, The University of Auckland, New Zealand, 2007.
- [129] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.
- [130] Maryam Hedayati, Shima Sum, Seyed Reza Hosseini, Mahbobeh Faramarzi, and Samaneh Pourhadi. Investigating the effect of physical games on the memory and attention of the elderly in adult day-care centers in babol and amol. *Clinical interventions in aging*, pages 859–869, 2019.
- [131] Chaofan Wang, Zhanna Sarsenbayeva, Chu Luo, Jorge Goncalves, and Vasilis Kostakos. Improving wearable sensor data quality using context markers. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC ’19 Adjunct,

- page 598–601, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368698. doi: 10.1145/3341162.3349334. URL <https://doi.org/10.1145/3341162.3349334>.
- [132] Jonghyun Ryu. Psychophysical model for vibrotactile rendering in mobile devices. *Presence*, 19(4):364–387, 2010.
- [133] Massimo Bovenzi et al. Health effects of mechanical vibration. *G Ital Med Lav Ergon*, 27(1):58–64, 2005.
- [134] Helmut Seidel. Selected health risks caused by long-term, whole-body vibration. *American journal of industrial medicine*, 23(4):589–604, 1993.
- [135] Simon rockliff’s reed-solomon encoder/decoder. <http://www.eccpage.com/rs.c>.
- [136] Christian Rathgeb and Andreas Uhl. Statistical attack against fuzzy commitment scheme. *IET biometrics*, 1(2):94–104, 2012.
- [137] Jeroen Delvaux, Dawu Gu, Dries Schellekens, and Ingrid Verbauwhede. Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis. *IEEE TCAD*, 34(6), 2015.
- [138] Mana Al Reshan, Hang Liu, Chunqiang Hu, and Jiguo Yu. Mbpska: Multi-biometric and physiological signal-based key agreement for body area networks. *IEEE Access*, 7, 2019.
- [139] TI. Msp430fr596x, msp430fr594x mixed-signal microcontrollers datasheet, rev. g, 2012. <https://www.ti.com/lit/gpn/msp430fr5969>.
- [140] TI. Msp430fr599x, msp430fr596x mixed-signal microcontrollers datasheet, rev. c, 2016. <https://www.ti.com/lit/gpn/msp430fr5994>.
- [141] Giacomo de Meulenaer, François Gosset, François-Xavier Standaert, and Olivier Pereira. On the Energy Cost of Communication and Cryptography in Wireless Sensors Networks. In *IEEE WiMob*, pages 580–585, 2008.
- [142] Crossbow Technology Inc. Telosb mote platform datasheet, rev. b, 2012. https://www.willow.co.uk/TelosB_Datasheet.pdf.
- [143] TI. An092: Measuring bluetooth low energy power consumption, 2012.
- [144] Oscar Reparaz, Josep Balasch, and Ingrid Verbauwhede. Dude, is my code constant time? In *DATE*, pages 1697–1702. IEEE, 2017.
- [145] Gesine Hinterwälder, Amir Moradi, Michael Hutter, Peter Schwabe, and Christof Paar. Full-size high-security ecc implementation on msp430 microcontrollers. In *LATINCRYPT*, pages 31–47. Springer, 2014.

- [146] Sriram Cherukuri, Krishna K Venkatasubramanian, and Sandeep K S Gupta. Biosec: a biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. In *ICPP*, pages 432–439, 2003.
- [147] C. Hu, X. Cheng, F. Zhang, D. Wu, X. Liao, and D. Chen. OPFKA: Secure and efficient Ordered-Physiological-Feature-based key agreement for wireless Body Area Networks. In *INFOCOM*, 2013.
- [148] Krishna K Venkatasubramanian, Ayan Banerjee, and S Gupta. Plethysmogram-based secure inter-sensor communication in body area networks. In *IEEE MIL-COM*, 2008.
- [149] Krishna K. Venkatasubramanian, Ayan Banerjee, and Sandeep Kumar S. Gupta. PSKA: Usable and Secure Key Agreement Scheme for Body Area Networks. *IEEE T-ITB*, 14(1):60–68, 2010.
- [150] VERY. Manual vs. ota firmware updates for iot, 2019. <https://www.verytechology.com/iot-insights/manual-vs-ota-firmware-updates-for-iot>.
- [151] S. Abhishek Anand and Nitesh Saxena. A sound for a sound: Mitigating acoustic side channel attacks on password keystrokes with active sounds. In *Financial Cryptography and Data Security*, pages 346–364, 2017. ISBN 978-3-662-54970-4.
- [152] Phyllis A Guze. Using technology to meet the challenges of medical education. *Transactions of the American clinical and climatological association*, 126: 260, 2015.
- [153] MDM Healthcare. Complete patient education technology: Improve engagement & enhance experiences, 2022. <https://www.mdmcommercial.com/healthcare-technology/patient-education/>.
- [154] Alex Campolo, Madelyn Rose Sanfilippo, Meredith Whittaker, and Kate Crawford. Ai now 2017 report. 2017.
- [155] Jenny Waycott, Greg Wadley, Stefan Schutt, Arthur Stabolidis, and Reeva Lederman. The challenge of technology research in sensitive settings: Case studies in 'ensitive hci'. In *Proceedings of the Annual Meeting of the Australian Special Interest Group for Computer Human Interaction*, pages 240–249, 2015.
- [156] Carmen Camara, Pedro Peris-Lopez, and Juan E Tapiador. Security and privacy issues in implantable medical devices: A comprehensive survey. *Journalpacemakerconsent of biomedical informatics*, 55:272–289, 2015.

- [157] Ewa Luger and Tom Rodden. An informed view on consent for ubicomp. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 529–538, 2013.
- [158] Matthias Fassel, Lea Theresa Gröber, and Katharina Krombholz. Stop the consent theater. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.
- [159] Jane Im, Jill Dimond, Melody Berton, Una Lee, Katherine Mustelier, Mark S Ackerman, and Eric Gilbert. Yes: Affirmative consent as a theoretical framework for understanding and imagining social platforms. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–18, 2021.
- [160] Ewa Luger, Stuart Moran, and Tom Rodden. Consent for all: revealing the hidden complexity of terms and conditions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 2687–2696, 2013.
- [161] Ping Guo, Kim Watts, and Heather Wharrad. An integrative review of the impact of mobile technologies used by healthcare professionals to support education and practice. *Nursing Open*, 3(2):66–78, 2016.
- [162] M Lumini, Fátima Araújo, and Teresa Martins. The role of educational technology in caregiving. *Caregiving and home care*, 11:179–01, 2018.
- [163] Francine Gemperle, Chris Kasabach, John Stivorc, Malcolm Bauer, and Richard Martin. Design for wearability. In *digest of papers. Second international symposium on wearable computers (cat. No. 98EX215)*, pages 116–122. IEEE, 1998.
- [164] Clint Zeagler. Where to wear it: functional, technical, and social considerations in on-body location for wearable technology 20 years of designing for wearability. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 150–157, 2017.
- [165] Tamara Denning, Daniel B Kramer, Batya Friedman, Matthew R Reynolds, Brian Gill, and Tadayoshi Kohno. Cps: Beyond usability: Applying value sensitive design based methods to investigate domain characteristics for security for implantable cardiac devices. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 426–435, 2014.