



COMPROMISING ELECTROMAGNETIC EMANATIONS:  
SIDE-CHANNEL LEAKAGES IN EMBEDDED DEVICES

PIERRE AYOUB

*Software and System Security (S3) Group*  
EURECOM

Thesis submitted to  
*Sorbonne Université*  
*École Doctorale Informatique, Télécommunications et Électronique*  
*(EDITE de Paris, ED130)*

Publicly defended at EURECOM the 03/12/2024 in front of

Director	<i>Pr. Aurélien Francillon (EURECOM)</i>
Supervisor	<i>CR. Clémentine Maurice (CNRS)</i>
Supervisor	<i>MdC. Romain Cayre (INSA/LAAS)</i>
Reviewer	<i>Pr. François-Xavier Standaert (Pr. UC Louvain)</i>
Reviewer	<i>Pr. Guy Gogniat (Univ. Bretagne Sud)</i>
Jury President	<i>Pr. Raymond Knopp (EURECOM)</i>
Jury	<i>Pr. Colin O'Flynn (Dalhousie Univ. / NewAE Technology)</i>
Jury	<i>Ph.D. Jose Lopes Esteves (ANSSI)</i>
Guest	<i>MdC. Robin Gerzaguët (ENSSAT/IRISA)</i>
Guest	<i>Pr. Maxime Pelcat (INSA/IETR)</i>

For the award of the degree of  
*Philosophiæ Doctor (Ph.D.)*

March 2025



## ABSTRACT

**M**ODERN electronic devices are increasingly interconnected and integrated. Communications security mainly relies on cryptographic algorithms which ensure a mathematically guaranteed level of confidentiality. However, when algorithms are executed by the hardware, they inevitably interact with their physical environment. This can lead to sensitive information being inferred from physical measurements. Attacks exploiting these measurements instead of the main channel are known as *side-channel attacks*, leveraging an unintentional relation between a physical quantity and the secret.

In this thesis, we focus our work on new security risks impacting microcontrollers linked to unexpected interactions between heterogeneous digital and analog embedded modules. In particular, we analyze threats related to cross-layer interactions that can be exploited through electromagnetic side-channel analysis and propose two main contributions. First, despite a flourishing literature on electromagnetic side channels, the modulation of leaked signals remains a complex phenomenon which is not fully understood. In this context, the first major contribution of our work aims at enhance understanding of how digital activity modulate leaked electromagnetic signals for both offensive and defensive applications. More precisely, electromagnetic side channels typically focus on the amplitude of leaked signals, neglecting the potential interest of other modulation types from a security perspective. Our first contribution, *PhaseSCA*, uncovers unintended phase modulation in leaked signals as a novel source of side-channel. Second, the applicability of newly discovered side-channel attacks regarding modern communication protocols is not systematically evaluated. An illustration of this is the *Screaming Channels* attack, which exploits a phenomenon of intermodulation between the leakage from a digital activity and the carrier of a radio transceiver in mixed-signal chips. Modern protocols only enable the radio transceiver for a short duration, introducing a serious limitation since Screaming Channels exploits leakage broadcasted through the radio transceiver. Our second contribution explores the impact of this threat on Bluetooth Low Energy protocol. We highlight how an attacker can manipulate the protocol parameters through traffic injection, forcing a victim to transmit during sensitive operations, demonstrating the threat introduced by Screaming Channels for the Internet of Things ecosystem.

Overall, this work underscores the need to systematically analyze the unexpected interactions between hardware and software to design secure chips and protocols, as well as highlight emerging security threats and their implications for wireless communications.



## RÉSUMÉ

Les objets électroniques sont de plus en plus interconnectés. La sécurité des communications est assurée par des algorithmes cryptographiques, interagissant inévitablement avec leur environnement lorsqu'ils sont exécutés par du matériel. Un attaquant inférant des informations sensibles depuis des mesures physiques met en œuvre une *attaque par canal auxiliaire*, se basant sur une relation non intentionnelle entre une grandeur physique et l'information secrète.

Cette thèse se concentre sur les nouveaux risques de sécurité impactant les microcontrôleurs liés aux interactions inattendues entre des modules embarqués hétérogènes. En particulier, nous analysons les risques liés aux interactions entre couches pouvant être exploités *via* une analyse par canal auxiliaire d'émanation d'onde électromagnétique, puis proposons deux contributions principales. Premièrement, malgré une abondante littérature sur les attaques exploitant des fuites électromagnétiques, leur modulation reste un phénomène complexe qui n'est pas entièrement compris. Dans ce contexte, notre première contribution a pour but d'améliorer la compréhension de l'impact de l'activité numérique sur la modulation des fuites de signaux électromagnétiques, dans une visée offensive et défensive. Plus précisément, une attaque par canal auxiliaire se concentre généralement sur l'amplitude des signaux, négligeant l'impact potentiel des autres types de modulation sur la sécurité. Notre première contribution, *PhaseSCA*, démontre comment la modulation de phase non intentionnelle de signaux fuités se présente comme une nouvelle source de canal auxiliaire. Deuxièmement, l'impact des dernières attaques par canal auxiliaire sur les protocoles de communication modernes n'est pas systématiquement évalué. Une illustration en est l'attaque *Screaming Channels*, qui exploite un phénomène d'intermodulation dans une puce à signaux mixtes entre la fuite résultante de l'activité numérique et la porteuse du transmetteur radio. Les protocoles modernes n'activant ce dernier que pendant une courte période, cela impose une importante limite à l'exploitation de la fuite émise par la radio. Notre seconde contribution explore l'impact de cette attaque sur le Bluetooth Low Energy. Démontrant comment un attaquant peut manipuler les paramètres du protocole *via* de l'injection de paquet pour forcer une victime à transmettre pendant l'opération sensible, nous soulignons la menace introduite par cette attaque pour l'écosystème de l'Internet des Objets.

Dans l'ensemble, ce travail souligne le besoin d'une analyse systématique des interactions inattendues entre matériel et logiciel afin de concevoir des puces et protocoles sécurisés, ainsi que les risques de sécurité émergents pour le futur des communications sans fil.



## PUBLICATIONS

This thesis manuscript builds upon published research papers, while contributing additional findings and insights.

The first article [Ayo+24a] has been published at the ACSAC conference. It explores the application of *Screaming Channels*, a side-channel attack exploiting electromagnetic emanations over long distances, on the Bluetooth Low Energy protocol. The second article [Ayo+24b] has been published at the TCHES journal. Titled *PhaseSCA*, this work uncovers phase-modulated electromagnetic side channel leakage and introduce a novel method to exploit them.

Another article [AM21] was published in the EuroSec workshop during the first year of my Ph.D. It is based on my master's thesis work which involved reproducing *Spectre*, a microarchitectural attack, on real and simulated hardware. Since this work have a dedicated master thesis manuscript untitled *Simulating Transient Execution Attacks with gem5*, its content will not be part of this thesis.

- [Ayo+24a] Pierre Ayoub, Romain Cayre, Aurélien Francillon, and Clémentine Maurice. "BlueScream: Screaming Channels on Bluetooth Low Energy." In: *40th Annual Computer Security Applications Conference (ACSAC '24)*. Waikiki, Honolulu, Hawaii, United States, Dec. 2024. URL: <https://hal.science/hal-04725668> (cit. on p. vii).
- [Ayo+24b] Pierre Ayoub, Aurélien Hernandez, Romain Cayre, Aurélien Francillon, and Clémentine Maurice. "PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels." In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) 2025.1* (Dec. 2024), pp. 392–419. DOI: [10.46586/tches.v2025.i1.392-419](https://doi.org/10.46586/tches.v2025.i1.392-419). URL: <https://tches.iacr.org/index.php/TCHES/article/view/11934> (cit. on p. vii).
- [AM21] Pierre Ayoub and Clémentine Maurice. "Reproducing Spectre Attack with gem5: How To Do It Right?" In: *Proceedings of the 14th European Workshop on Systems Security*. EuroSec '21. Online, United Kingdom: Association for Computing Machinery, Apr. 2021, pp. 15–20. ISBN: 9781450383370. DOI: [10.1145/3447852.3458715](https://doi.org/10.1145/3447852.3458715). URL: <https://doi.org/10.1145/3447852.3458715> (cit. on p. vii).





## ACKNOWLEDGMENTS

**T**HIS manuscript concludes the last 4 years of my life. From the beginning of this adventure, I would have never imagined all the difficulties I was going through. Only the people closest to me know that this journey, while being the most difficult project of my life until then, is also one of my proudest achievements. I had the opportunity to work on fascinating topics, with a great degree of liberty when choosing my directions. Without the support I had during those past years, I would never have been able to conclude this thesis. For all those reasons, I would like to sincerely thank all the following people.

First, I want to express my gratitude to you *Aurélien Francillon*, for your teaching about radio security and for supervising me during all those years while putting your trust in my work. Back to a memorable conversation in 2023, without your support to persist in the thesis, I would probably have not been able to succeed toward the end of the Ph.D. Thank you *Clémentine Maurice*, for your wise advice, your “LaTeX-fu” and for supervising me since the beginning of my master thesis — such a long time from now! Finally, last but not least, thank you *Romain Cayre* for your supervising, your technical knowledge about IoT, and your kind support. It was a great pleasure and an enriching experience to collaborate with you during all my work. Thanks to the institutions that made my Ph.D. possible and to their directors, EURECOM led by *David Gesbert*, and the EDITE from Sorbonne Université.

Second, I want to thank my jury and reviewers: *Raymond Knopp* for the direction of the jury, *Francois-Xavier Standaert* for his expertise in statistics, *Guy Gogniat* for his deep review of my manuscript, *José Lopes Esteve* for his knowledge about signals and EMC, *Colin O’Flynn* for our exchanges about our respective work, *Robin Gerzaguét* for its surprising quick implementation of one of our result, and *Maxime Pelcat* for his comments. Thank you for your time, for your reading of this long manuscript, and for your corrections — it was an honor to present you my work and have new perspectives thanks to our discussions.

As important, I would like to thank all of my colleagues from EURECOM and the S3 team, as well as the people encountered during travels and conferences. More especially, I would like to extend a warm thanks to you *Aurélien Hernandez*, for all those passionate discussions we had about hardware, radio, life, and your support. Collaborating with you was one of the best memories from this journey, and I am really happy we did this together. I also say a big thanks to *Romain Malmain* and *Simon Autechaud*, who participated greatly in making the environment a nice place to be.

Dear *Maëlle*, thank you for all of the last 2 years. You have been the only person staying up all nights before submission deadlines to help me do the best I could. Above all, thank you for all your love and kindness during those years. Dear family, my parents, my brothers, I would also say a huge thank you – every one of you participated more in this thesis than you could imagine. I would mention a special thanks to my father, who put a computer in my hands before I knew how to ride a bicycle – I do both well today! –, without who my passion for computer science would have not evolved like it is today.

Thank you my long-time friends, *Corentin* and *Corentin*, *Théo*, *Gabriel*, *Damien*, *Sébastien*, *Valentin* and *Juliette*, on whom I always have been able to rely on during the last years.

Since this will never be enough,

*Thank you all*

# CONTENTS

1	Introduction	1
1.1	A Brief History	1
1.1.1	Radio communications	2
1.1.2	Computer communications	3
1.2	Modern Secure Communications	4
1.2.1	Cryptography for Secure Communications	4
1.2.2	Side-channel attacks	4
1.3	On Interdisciplinarity	6
1.3.1	Emerging Threats at the Boundaries	6
1.3.2	Convergence of Complementary Disciplines	7
1.4	Contributions	7
1.4.1	Academic Contributions	9
1.4.2	Open-Source Tools & Data	10
1.4.3	Outline	10
<b>I</b>	<b>Overview of Electromagnetic Side Channels</b>	
2	Background	15
2.1	Radio Communication	15
2.1.1	Signal Representation	15
2.1.2	Radio Architecture	18
2.1.3	Measurement Equipment for Side Channels	18
2.2	Electromagnetic Compatibility (EMC)	19
2.2.1	Interference	20
2.2.2	Emanation	20
2.2.3	Electric Field and Magnetic Field	21
2.2.4	Near-Field and Far-Field	24
2.2.5	Coupling	25
2.2.6	Non-linearities	30
2.2.7	Crosstalk	33
2.3	Advanced Encryption Standard (AES)	34
3	State of the Art	37
3.1	Fundamental Terminology	37
3.1.1	Introduction of Security Terms	37
3.1.2	The Multiple Definitions of “Leak”	38
3.2	Compromising Emanations and EMSEC	38
3.2.1	Sensitivity	39
3.2.2	Directness	40
3.2.3	Intentionality	44
3.2.4	Activeness	44
3.2.5	Escape Medium	45
3.3	Electromagnetic Attacks	45
3.3.1	NSA Code Names	46

3.3.2	Illumination (Re-Emission) . . . . .	48
3.3.3	Soft TEMPEST . . . . .	50
3.3.4	Van Eck Phreaking . . . . .	51
3.3.5	Jamming . . . . .	52
3.4	Side-Channel Attacks . . . . .	53
3.4.1	Dependency . . . . .	55
3.4.2	Media . . . . .	55
3.4.3	Overview . . . . .	56
3.4.4	Multi-Channel Attacks . . . . .	59
3.4.5	Evaluation Metrics . . . . .	59
3.5	Electromagnetic Side-Channel Attacks . . . . .	60
3.5.1	Distant Attacks . . . . .	61
3.5.2	Leakage Detection . . . . .	63
ii	<b>BlueScream: Screaming Channels on Bluetooth Low Energy</b>	
4	Motivations . . . . .	69
4.1	Potential Impact . . . . .	69
4.2	Contribution . . . . .	70
4.3	Bluetooth Low Energy . . . . .	71
4.4	Related Work . . . . .	76
4.4.1	Prior Work . . . . .	76
4.4.2	Parallel Work . . . . .	77
5	Attacking Bluetooth Low Energy . . . . .	79
5.1	Threat Model . . . . .	79
5.2	Attack Overview . . . . .	80
5.3	Experimental Setup . . . . .	82
5.4	Bluetooth Low Energy Manipulation . . . . .	86
5.4.1	Challenges . . . . .	86
5.4.2	Methodology . . . . .	87
5.4.3	Evaluation . . . . .	90
5.5	Screaming Channel Attack . . . . .	93
5.5.1	Side-channel attack on AES . . . . .	93
5.5.2	Profiled Correlation Attack . . . . .	94
5.5.3	Evaluation on Firmware $F_{instru}$ . . . . .	95
5.5.4	Evaluation on Firmware $F_{default}$ . . . . .	96
5.5.5	Impact of Noise on the Profile . . . . .	96
6	Observations and Conclusions . . . . .	99
6.1	Leakage Characterization . . . . .	99
6.1.1	Profile Reuse . . . . .	99
6.1.2	Leaking Frequencies . . . . .	100
6.1.3	Hardware Component Impacting the Leakage . . . . .	103
6.2	Discussion . . . . .	103
6.2.1	Protocol Manipulation . . . . .	103
6.2.2	Attack Deployment and Impact . . . . .	104
6.3	Countermeasures . . . . .	105
6.4	Conclusion . . . . .	106

<b>iii PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels</b>	
<b>7</b>	<b>Motivations . . . . . 109</b>
7.1	Discovery . . . . . 111
7.1.1	The First Experiments . . . . . 111
7.1.2	Potential Impact . . . . . 112
7.2	Contribution . . . . . 113
7.3	Signal Jitter and Phase Shift . . . . . 115
7.4	Related Work . . . . . 115
7.4.1	Foundational Work . . . . . 116
7.4.2	Recent Work . . . . . 117
7.4.3	Parallel Work . . . . . 118
7.4.4	Proxy Measurement . . . . . 118
<b>8</b>	<b>Phase Modulated Side Channels . . . . . 121</b>
8.1	Threat Model . . . . . 121
8.2	Methodology . . . . . 121
8.2.1	Side-Channel Trace . . . . . 122
8.2.2	Side-Channel Attack . . . . . 125
8.2.3	Generalization . . . . . 128
8.2.4	Reproduction . . . . . 128
8.3	Experimental Setup . . . . . 131
8.3.1	Side-Channel Attack . . . . . 131
8.3.2	Jitter and Phase Shift Reproduction . . . . . 132
8.4	Evaluation . . . . . 134
8.4.1	Identifying Phase-Modulated Leakage . . . . . 134
8.4.2	Side-Channel Attack using Phase Shift . . . . . 137
8.4.3	Jitter and Phase Shift Reproduction . . . . . 141
<b>9</b>	<b>Observations and Conclusion . . . . . 145</b>
9.1	Discussion . . . . . 145
9.2	Countermeasures . . . . . 147
9.3	Conclusion . . . . . 147
<b>iv Perspectives</b>	
<b>10</b>	<b>Countermeasures . . . . . 151</b>
10.1	Taxonomy . . . . . 151
10.1.1	Physical . . . . . 151
10.1.2	Cryptography . . . . . 153
10.1.3	Specification . . . . . 154
10.1.4	Software . . . . . 155
10.2	Lessons Learned . . . . . 155
<b>11</b>	<b>Concluding Remarks . . . . . 159</b>
11.1	Limitations . . . . . 159
11.2	Future Work . . . . . 160
11.2.1	Continuing Our Work . . . . . 160
11.2.2	Improving Side Channels . . . . . 161
11.3	Conclusion . . . . . 162

Bibliography  
Index

193

## LIST OF FIGURES

Figure 1.1	Chappe Optical Telegraph from Louis Figuiet (CC-PD-Mark) [Fig22] . . . . .	2
Figure 1.2	Block diagram of a Mixed-Signal System-on-Chip (MSoC). . . . .	8
Figure 2.1	Representation of a real-valued signal. A difference in amplitude or in period ( <i>i.e.</i> , inverse of frequency) are delimited using dashes, while a difference in instantaneous phase is illustrated using a shift of $\pi$ for the orange signal. . . . .	16
Figure 2.2	Representation of a complex-valued signal ( <i>i.e.</i> , analytic representation). . . . .	16
Figure 2.3	Depending on the expression unit (which depends on the measurement tool), a difference in two phase values will be expressed or represented in different ways. . . . .	17
Figure 2.4	Radio receiver direct-conversion architecture. . . . .	18
Figure 2.5	Different forms and media where an unintentional emanations may propagate through. . . . .	21
Figure 2.6	A current (red) flowing across a conductor will induce a magnetic flux (blue) in its vicinity. . . . .	23
Figure 3.1	A difference in the secret input of an <b>integrated circuit (IC)</b> will produce a difference in its physical emanation – being compromising. . . . .	39
Figure 3.2	Illustration of a carrier being modulated in amplitude by a message (red) signal. As a result, the amplitude of the modulated signal is varying according to the value (instantaneous amplitude) of the message signal. . . . .	41
Figure 3.3	Illustration of a carrier being modulated in phase by a message (red) signal. As a result, the frequency of the modulated signal is varying according to the variation of the value (instantaneous amplitude) of the message signal. . . . .	42
Figure 3.4	Example of a non-profiled correlation based side-channel attack. . . . .	57
Figure 3.5	Coupling path (in red) inside a <b>system-on-chip</b> vulnerable to Screaming Channels as studied by <i>Camurati et al.</i> [Cam+18]. The conventional <b>compromising emanation</b> are still emitted by the computational units ( <i>e.g.</i> , the CPU and the CRYPTO blocks), but also additionally by the radio transceiver during a radio transmission. . . . .	61

Figure 3.6	Comparison of the consequences of a Conventional Side-Channel attack <i>vs.</i> a Screaming Channel attack. As a result of the Screaming Channel leakage, the <a href="#">compromising emanation</a> (in red) is upconverted to the frequency of the <i>internal intended RF carrier</i> and broadcasted at a higher distance. . . . .	62
Figure 4.1	Session key derivation in BLE. The ciphertext $C$ will be sent over the air to transmit the message $M$ . . . . .	73
Figure 4.2	Standard BLE communication divided in CE according to Hop Interval $H$ . . . . .	74
Figure 5.1	Attack overview. . . . .	81
Figure 5.2	Experimental setup . . . . .	83
Figure 5.3	Leakage characterization setup using two synchronized USRPs used in Section 5.4. . . . .	84
Figure 5.4	Collection setup using the directional antenna. . . . .	85
Figure 5.5	One of the challenge of Screaming Channels is that a radio transmission must happen at the same time of a sensitive operation to have a <a href="#">compromising emanation</a> . . . . .	86
Figure 5.6	A BLE connection is divided into successive <a href="#">connection event (CE)</a> , that the attacker can use to target a specific moment in time. . . . .	87
Figure 5.7	The attacker can increase the number of packets inside one <a href="#">connection event (CE)</a> by set the <a href="#">More Data (MD)</a> bit to 1. This increase the probability of having a transmission during a sensitive operation. . . . .	88
Figure 5.8	The attacker can increase the duration of a <a href="#">connection event (CE)</a> by increasing the Hop Interval value. This decrease the number of radio reconfiguration for a specific time window, thus increasing the probability of having a transmission during a sensitive operation. If the <a href="#">More Data (MD)</a> bit is set to 1, even more packets can be transmitted during a CE (gray shaded). . . . .	88
Figure 5.9	By sending multiple requests concurrently ( <i>i.e.</i> , interleaving procedures), the attacker can make the victim transmit an answer during a sensitive operation. The larger will be the answer by the victim device, the higher will be the probability to have a radio transmission during a sensitive operation. . . . .	89
Figure 5.10	Comparison of system activity leakage duration based on procedure interleaving method. . . . .	91
Figure 5.11	Cryptographic leakage detection. . . . .	92
Figure 5.12	<a href="#">Bluetooth Low Energy (BLE)</a> session key derivation. The <a href="#">long term key (LTK)</a> is the input key, the <a href="#">session key diversifier (SKD)</a> is the input plaintext, and the <a href="#">session key (SK)</a> is the output ciphertext. . . . .	93





Figure 5.13	Time diversity allows to improve the <a href="#">signal-to-noise ratio (SNR)</a> of measurements by averaging a high number of traces collected under the same conditions. However, for an attack under realistic conditions, this method is no longer suitable due to the lack of control for the cryptographic input. . . . .	94
Figure 5.14	Comparison of the key rank over the number of traces for $A_7$ ( <a href="#">anechoic</a> ) and $A_9$ (office) using $F_{default}$ . $A_9$ is converging toward a limit due to the noise floor.	97
Figure 5.15	Correlation on amplitude for $A_7$ during a radio transmission with GFSK at $f_{carrier} + 2 * f_{clock}$ Hz.	97
Figure 6.1	Failed profile reuse between two different conditions for the $F_{default}$ firmware. The green trace is the average of the profile while the red trace is the average of the attack traces. . . . .	100
Figure 6.2	Third-order intermodulation product at 2.496 GHz between the 32 MHz CPU sub-clock and the 2.4 GHz carrier. . . . .	101
Figure 6.3	Correlations with instruction cache disabled (top) and enabled (bottom). Enabling instruction cache still allows finding correlations, but create interruptions in the leakage.	102
Figure 6.4	The CPU (red) can fetch instructions from either the RAM (blue), the I-Cache (purple) or the Flash (green). [ <a href="#">Sem21</a> , p. 24] . . . . .	102
Figure 7.1	Demodulated AES leak signal from Screaming Channels frequencies in the <a href="#">far-field</a> (2.5 GHz).	110
Figure 7.2	Hypothesis about a possible phase modulated Screaming Channel leakage. A hypothetical coupling path between the CPU and the <a href="#">voltage-controlled oscillator (VCO)</a> of the RF <a href="#">phase-locked loop (PLL)</a> would imply a phase-modulated output of the radio transceiver by the activity of the CPU. . . . .	111
Figure 7.3	Summary of frequency ranges containing a phase-modulated leakage by the CPU activity. The CPU clock frequency, its harmonics, and the intermodulated result in the Screaming Channel leakage contains phase-modulated leakage, but not the RF carrier. . . . .	112
Figure 7.4	Computation of an "amplitude trace" — performing an amplitude demodulation of the complex-valued signal. . . . .	116
Figure 7.5	State of the art for proxy measurements. While it is widely known that <a href="#">EM</a> amplitude is a proxy measurement for power analysis in side channels, our work and other recent work explore how <a href="#">EM</a> phase and signal jitter relates to power analysis.	119
Figure 8.1	Illustration of the 3 principal steps of a phase shift trace computation (detailed in <a href="#">Section 8.2.1</a> ).	123

Figure 8.2	The instantaneous phase is computed by taking the argument of each complex samples. The result of this process corresponds to the first plot of Fig. 8.1. . . . .	124
Figure 8.3	Amplitude (top) and phase (bottom) correlation-based side-channel attacks, independently of each others. . . . .	125
Figure 8.4	Waterfall illustrating filters isolating amplitude and phase shift leakage. A low-pass filter is used to isolate the phase-modulated leakage, while a high-pass filter is used to isolate the amplitude-modulated leakage. . . . .	126
Figure 8.5	Multi-channel attack recombining amplitude and phase attack results into a single attack. . . . .	127
Figure 8.6	Hypothetical jitter source from processor activity.	129
Figure 8.7	Hardware experimental setup for side-channel evaluation. . . . .	131
Figure 8.8	Simplified view of the STM32F103RB internal clocking circuit. . . . .	132
Figure 8.9	Experimental setup . . . . .	133
Figure 8.10	AES influence on the continuous instantaneous phase (CIP) of signals. . . . .	135
Figure 8.11	AES trace in amplitude (left) and phase shift (right). Captured from an nRF52832 using a <a href="#">near-field</a> probe connected to an <a href="#">software-defined radio (SDR)</a> tuned at 138 MHz. . . . .	136
Figure 8.12	Correlation coefficients ( $\rho$ ) for POIs on phase shift for the nRF52. . . . .	137
Figure 8.13	Performance over number of traces for profiled attacks. The smaller, the better. . . . .	139
Figure 8.14	Performance over number of traces for non-profiled attacks. The smaller, the better. . . . .	140
Figure 8.15	STM32F1 internal clock signal measurement under various CPU states, clock configurations, and overconsumption events. . . . .	143

## LIST OF TABLES

Table 5.1	Firmware used during this work. . . . .	84
Table 5.2	Attack results using $F_{\text{instru}}$ . . . . .	95
Table 5.3	Attack results using $F_{\text{default}}$ . . . . .	96
Table 5.4	Means of <a href="#">Pearson correlation coefficient (PCC)</a> ( $\bar{\rho}$ ) and standard deviation ( $\bar{\sigma}$ ) during profiles creation. . . . .	96
Table 8.1	Multiple target devices and SoCs evaluated in the side-channel analysis. . . . .	131

Table 8.2	Summary of results for evaluated SoCs in the side-channel analysis. Legends:  successful exploitation,  unsuccessful exploitation . . . . .	136
-----------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

## ACRONYMS

ADC Analog-To-Digital Converter . . . . .	3, 18, 19
AES Advanced Encryption Standard . . . . .	15, 34, 73
AM Amplitude Modulation . . . . .	51, 64
AMC Automatic Modulation Classification . . . . .	64
ASIC Application-Specific Integrated Circuit . . . . .	xxiv
BLE Bluetooth Low Energy xvi, 7, 69–71, 74–80, 82, 84, 85, 87, 90–94, 99, 104–106, 110, 147, 153, 159, 193	
BR/EDR Basic Rate / Enhanced Data Rate . . . . .	72, 76
CE Compromising Emanation . . . xv, xvi, 15, 20, 25, 26, 29, 40, 44, 45, 47–52, 61, 62, 86, 151–155, 159, 160	
CE Connection Event . . . . .	xvi, 74, 75, 87–90, 104, 106, 193
CIP Continuous Instantaneous Phase . . . . .	122, 124, 134, 135
CMOS Complementary Metal–oxide–semiconductor 27, 43, 100, 193	
CO Cryptographic Operation . 53, 57, 61, 63, 71, 77, 78, 86, 88–92, 95, 100, 106, 113, 155, 193	
CPA Correlation Power Analysis . . . . .	55, 57
DEMA Differential Electromagnetic Attack . . . . .	60
DLL Delay Locked Loop . . . . .	117
DPA Differential Power Analysis . . . . .	34, 56, 60, 147
DSP Digital Signal Processing . . . . .	3, 15
DUT Device Under Test . . . . .	19, 20
EM Electromagnetic . . . xvii, xxi–xxiii, 18, 20, 33, 37–39, 42, 45, 49, 51, 60–62, 64, 69, 70, 77, 78, 93, 105, 109, 112, 113, 116–119, 121, 126, 129, 130, 135, 145, 147, 148, 151, 161	
EMC Electromagnetic Compatibility xxi, xxii, 15, 19, 20, 33, 34, 40, 43, 45, 46, 100, 146, 147, 151, 152, 156, 159, 161, 193	
EMFA Electromagnetic Fault Attack . . . . .	45
EMFI Electromagnetic Fault Injection . . . . .	xxiii
EMI Electromagnetic Interference . . . . .	
EMR Electromagnetic Radiation . xxii, 19–21, 24, 28, 35, 44, 50–52, 56, 59–61, 63, 82, 91, 100, 114, 116, 118, 124, 131, 132, 134, 145, 147, 151, 152, 156	
EMSEC Emission Security . . . . .	xxii, 39, 40, 46
FCA Fault Correlation Analysis . . . . .	193

FF Far-Field	xvii, xxii, 21, 24, 25, 28, 29, 35, 61, 70, 78, 83, 84, 91–93, 96, 110, 160
FH Frequency Hopping	75, 87, 193
FI Fault Injection	119
FM Frequency Modulation	64
FT Fourier Transform	117
GFSK Gaussian Frequency-Shift Keying	193,, <i>Glossary: Gaussian Frequency-Shift Keying</i>
GPIO General-Purpose Input/output	130, 134, 141, 142
HD Hamming Distance	, <i>Glossary: Hamming Distance</i>
HW Hamming Weight	, <i>Glossary: Hamming Weight</i>
I-Cache Instruction Cache	103
I/Q In-Phase And Quadrature	17, 19
IC Integrated Circuit	xv, xxiv, 27, 29, 39, 50
IEMI Intentional Electromagnetic Interference	20, 45, 48, 52, 54,, <i>Glossary: Intentional Electromagnetic Interference</i>
IMD Intermodulation Distortion	32
LNA Low-Noise Amplifier	xxiv, 18, 83, 84, 96, 116, 193
LTK Long Term Key	xvi, 70–73, 78–80, 82, 93, 106, 193
MCFA Multi-Channel Fusion Attack	59
MCU Microcontroller Unit	xxiv
MD More Data	xvi, 74, 75, 82, 88–90, 193
MIMO Multiple-Input Multiple-Output	116
MSoC Mixed-Signal System-On-Chip	42, 62, 193, <i>Glossary: System-on-Chip</i>
NF Near-Field	xviii, 21, 24, 25, 27, 29, 34, 35, 61, 69, 78, 83, 84, 91–93, 109, 112, 118, 132, 136, 147, 160
OOK On-Off Keying	51
PCB Printed Circuit Board	50, 64, 152
PCC Pearson Correlation Coefficient	xviii, 58, 59, 96, 97, 128, 137
PDU Protocol Data Unit	74, 82, 88–90, 104
PGE Partial Guessing Entropy	60, 94, 95, 193
PIM Passive Intermodulation	32
PLL Phase-Locked Loop	xvii, 111, 112, 118, 129, 132, 133, 141, 147
PM Phase Modulation	64
POI Point Of Interest	58, 59, 97, 137
RF Radio-Frequency	xxiii, xxiv, 15, 20, 42, 48–50, 54, 56, 62, 64, 76, 83, 100, 101, 109, 111
RFFE Radio-Frequency Front-End	xxiv, 3,, <i>Glossary: Radio-Frequency Front-End</i>

RX Radio Reception .....	24, 44, 88, 89
SDR Software-Defined Radio	xviii, 3, 7, 10, 19, 52, 53, 63, 75, 78, 80, 83, 104, 105, 111–113, 115, 118, 122, 132, 136, 145, 147, 161, 193
SEMA Simple Electromagnetic Attack .....	60
SK Session Key .....	xvi, 73, 74, 89, 90, 93
SKD Session Key Diversifier .....	xvi, 73, 78, 80, 93, 193
SNR Signal-To-Noise Ratio .....	xvii, 77, 83, 94
SoC System-On-Chip ...	xv, 7, 10, 61, 113, 129, 152, 153, 160,, <i>Glossary:</i> <a href="#">System-on-Chip</a>
SPA Simple Power Analysis .....	34, 48, 55, 56, 60
STK Short-Term Key .....	72
TDC Time-To-Digital Converter .....	118
TX Radio Transmission	24, 44, 70, 71, 82, 86–92, 101, 103, 104, 106, 111, 112
VCO Voltage-Controlled Oscillator .....	xvii, 42, 43, 111, 112, 117, 147
XT Crosstalk .....	33, 40,, <i>Glossary:</i> <a href="#">Crosstalk</a>

## GLOSSARY

- Anechoic** Property of an environment which do not reflect acoustic or [electromagnetic \(EM\)](#) wave. For [electromagnetic compatibility \(EMC\)](#), it is often of the size of a box or a room — which is also a Faraday cage, blocking [EM](#) wave from outside. The largest anechoic “room” in the world is a hangar that contains entire aircraft. .... [xvii, 96, 97, 99, 100, 132](#)
- Baseband** Range of frequency containing a signal that has not been modulated. This signal contains the informative data that a receiver wants to decode after the demodulation process. [xxiii, xxiv, 18, 19, 32, 33, 38, 42, Glossary: Passband & Modulation](#)
- Cache** A cache memory is a faster but smaller memory storing data to accelerate future requests. It is typically connected to a bigger but slower memory (*e.g.*, a random-access memory) and placed at the closest of the processor. .... [103](#)
- Cache Hit** A data request that successfully retrieve a data from the cache, thus faster than a [cache miss](#). .... [xxi, 103](#)
- Cache Miss** A data request that failed to retrieve a data from the cache, thus slower than a [cache hit](#). .... [xxi, 103](#)
- Codebook** Physical document from the pre-digital era, used to store cryptographic codes in order to encrypt or decrypt a message.

- Coupling** Transmission of energy from a source (emitter, culprit) to a sink (receiver, victim) between two circuits. When unintentional, its consequences are called [electromagnetic interference](#) or [crosstalk](#). This phenomenon is studied, among others, in [electromagnetic compatibility](#). More information can be found in Section 2.2.5. . . . . xxiv, 20, 21, 25, 29, 45, 47, 49, 146, 160
- Crosstalk** From the [EMC](#) context, corresponds to the phenomenon of inductive or capacitive coupling — more information can be found in Section 3.2.1. From the [EMSEC](#) context, corresponds to the consequence of undesired propagation from a compromising signal into a legitimate channel — more information can be found in Section 2.2.7. . . . . xxii, 33, 34, 38, 40, 47, 151, *Acronyms: XT*
- Electromagnetic Interference** Specifically, an interference is a degradation in performance because of a disturbance. More generally, it is defined as the effect of a signal in one circuit creating undesired variations in another circuit. Interference refer to the consequences, *i.e.*, the effects seen from the receiver perspective. When the interference is from a compromising signal containing secret information, the term [crosstalk](#) is used in the [Emission Security \(EMSEC\)](#) context. More information can be found in Section 2.2.1. . . . . xxii, 20, 26, 151, *Acronyms: EMI*
- Electromagnetic Wave** Waves in the [electromagnetic](#) field carrying energy and propagating through space in the [far-field](#). More precisely, it corresponds to periodic changes in the electric (**E**) and the magnetic (**B**) fields which propagates point to point through space. [Electromagnetic radiation \(EMR\)](#) are composed of [electromagnetic](#) waves. . xxiv, 21, 24, 25, 28, 38, 39, 56, 152, 161
- Gaussian Frequency-Shift Keying** In FSK modulation, the instantaneous frequency of the carrier signal is shifted between discrete frequency values depending on the digital data symbols. In GFSK modulation, instead of directly using the digital data symbols, each symbol is filtered using a Gaussian filter to smooth transitions, to reduce sideband power and interference with neighboring channels. . . . . 72, *Acronyms: GFSK*
- Ground** Ideally, a reference point or surface of a zero-impedance and zero-potential for other voltage source [[ASo9](#), p. 115] [[Pauo6](#), p. 768]. In practice, when considering high-frequencies, all conductors have a parasitic impedance. It can be a *chassis ground*, *e.g.*, cases and enclosures, or an *earth ground*, or a *ground line*, *i.e.*, conductor acting as a return path for current. Its usual abbreviation in schematics is GND. . . . . 26

- Hamming Distance** The Hamming distance of two numbers is defined as the number of respective positions between the two numbers of their digits are different. . . . . 55, 57, 59, *Acronyms: HD*
- Hamming Weight** The Hamming weight of a number is defined as the number of occurrence of “1” considering the binary representation of a number. It is also known as population count or bit sum in low-level programming manuals. . . . . 35, 38, 55, 57, *Acronyms: HW*
- Illumination** **Electromagnetic** attack exploiting a cross-modulation of an external intended **radio-frequency (RF)** carrier because of a data-dependent phenomenon. First introduced by the NSA as *NONSTOP*, and then called *re-emission*. More information can be found in Section 3.3.2. . . . . xxiii, 43, 45, 48, 49, 62, 193
- Intentional Electromagnetic Interference** Attack technique using a **radio-frequency** signal injection for interfering or tampering with a system to defeat or bypass a security mechanism. It is used in **illumination** and **electromagnetic fault injection (EMFI)** attacks. . . . . 45, 48, 52, *Acronyms: IEMI*
- Internet of Things** Often abbreviated IoT, it designates electronic devices embedding sensors, processing and communication capabilities, connected to a private network or to the Internet. Examples of application area are smart homes, healthcare devices, industrial automation or security devices. . . . . 8, 9, 69, 71, 72, 79, 153, 157, 159, 160
- Key Derivation** Operation of deriving — *i.e.*, generating — a secret from another secret value. Typically, it is realized by a key derivation function (KDF) inside a cryptographic algorithm to derive a temporary key from a master key. . . . . 73
- LoRa** Modulation scheme using Chirp Spread Spectrum (CSS), where information is modulated as up-chirp, a signal whose frequency linearly increases over time, *i.e.*, performing a sweep. The sweep over the predefined bandwidth is cyclic (analog to a *modulo*). Different symbols are differentiated by adding a time shift to the chirp, which is detected when the chirp end-up a cycle. 51
- Modulation** Operation of modifying a parameter from the modulated signal, called the *carrier* signal, by the variations of the modulating signal, called the *message* signal. The frequency range of the carrier signal is called the **passband**, while it is called **baseband** for the message signal. . . . . 32, 43, 63, 109
- Passband** Range of frequency containing a signal that has been modulated into a carrier frequency. . . . . xxiii, xxiv, 38, 42, *Glossary: Baseband & Modulation*



- Principal Values** Principal values of a multi-valued function (*i.e.*, a function that has zero, one or more values in its range for one value of its domain) are the values along one branch of that function such as it becomes single-valued. . . . . 122, 125
- Radio-Frequency Front-End** First stage of a radio architecture after the antenna [Gra13, p. 2]. It is typically composed of a filter, a [low-noise amplifier \(LNA\)](#), and a mixer to perform the up or down conversion between the [baseband](#) frequency and the [passband](#) frequency. . . . . 3, *Acronyms: RFFE*
- Radio-Frequency Wave** [Electromagnetic wave](#) with a frequency comprised between 3 Hz and 3000 GHz. . . . . 15, 24, 28, 39
- Real-Time Operating System** Operating system designed for real-time computing, *i.e.*, where computation time is guaranteed with critical constraints. Examples of applications are for safety-critical embedded systems or timing-critical communication systems. . . . . 76
- Screaming Channels** Original publication of *Camurati et al.* [Cam+18] performing a side-channel attack at distance (several meters) retrieving an AES key. The underlying mechanism is a [coupling](#) phenomenon between the digital block (processing secret information) and the analog block broadcasting an [RF](#) modulated carrier. As such, the leakage from the digital block is up-converted, amplified and broadcast along the [RF](#) modulated carrier. More information can be found in Section 3.5.1.1. 43, 45
- System-on-Chip** Integrated circuit that embed several systems, *e.g.*, a processing unit, memory, input-output interface and its firmware. Depending on the usage, it can be designated as an [microcontroller unit \(MCU\)](#), an [application-specific integrated circuit \(ASIC\)](#), or other type of [IC](#). When integrating heterogeneous blocks, such as both digital processing units and analog [radio-frequency front-end \(RFFE\)](#) or analog sensor, the SoC is designated as a mixed-signal system-on-chip. . . , *Acronyms: SoC & MSoC*
- Telegraph** System to transmit messages across long distance using codes. . . . . 1
- Time-Division Duplex** Considering a half-duplex communication link, time-division duplex (TDD) enables full-duplex communication by using different time slots to separate emission and reception of two transceivers. . . . . 74



# 1

## INTRODUCTION

COMMUNICATION, or the act of exchanging information, is one of the most important requirements enabling humanity to progress. Today, people are accustomed to communicating over long distances through secure protocols. However, when the hardware implementing these protocols interacts with its environment, unexpected security threats emerge from the boundaries between abstraction layers. Given their nature, these security issues require interdisciplinary analysis. In this thesis, we analyze and address such threats by leveraging side-channel analysis of electromagnetic emanations. This section will first establish the context of our work through a brief historical introduction to secure communication systems. To conclude, we present our motivations, the goal of this work, and our contributions.

*Communication security is one of the greatest challenges for the 21st century.*

### 1.1 A BRIEF HISTORY

In ancient civilizations, exchanging arbitrary information over long distances was not possible. Nonetheless, several basic mechanisms were invented to transmit predefined messages, such as using smoke signals, horns, or carrier pigeons. Later, modern civilizations began to define *symbols*, creating a correspondence between a digit or letter and a signal transmitted using an optical or mechanical system. An example of a well-known invention is the French “*Sémaphore*” devised by Claude Chappe in 1794, also known as an optical **telegraph** (illustrated in Fig. 1.1). This system consisted of a tower built with movable wooden arms, where each position corresponded to a letter, word, or phrase decoded using a **codebook**. Long ago before the modern computer security context, one of the first hacking act targeted this Chappe’s telegraph [Did14]. More precisely, the financial investors François and Louis Blanc corrupted a few operators, in charge of embedding a code inside the messages from Paris to Tours, indicating the variations of Paris financial markets. Leveraging this *covert channel* strategy, they were informed about the future financial markets variations of Tours before anyone else. Later on in telecommunication evolution, with the discovery of electricity in the 18<sup>th</sup> century, the electrical telegraph was used instead of the optical telegraph. Another example is the Morse Code, invented by Samuel Morse and Alfred Vail in 1838, where digits and letters uniquely correspond to a series of pulses. One 19<sup>th</sup>-century example of such transmission equipment is the Heliograph, an optical

*Global communication was not conceivable before the discovery of radio-electricity.*

*The Blanc brothers are ones of the first hackers from the 19th-century.*

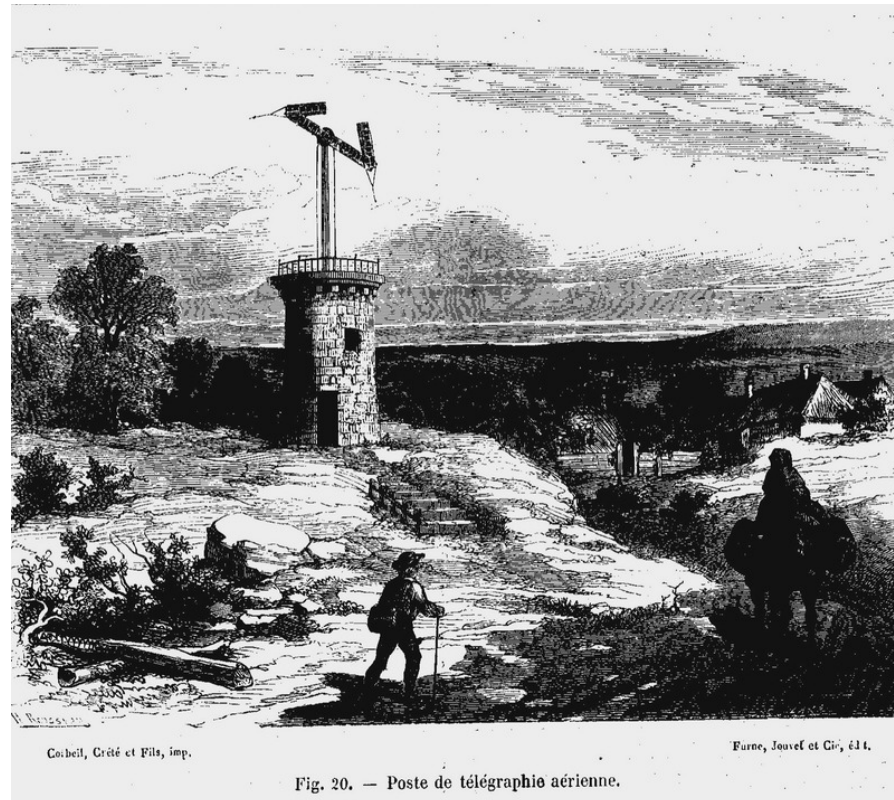


Figure 1.1: Chappe Optical Telegraph from Louis Figuier (CC-PD-Mark) [Fig22]

system equipped with a mirror that reflects sunlight, allowing the transmission of information up to a dozen kilometers for a naked-eye receiver. Despite its important limitation in sensitivity to weather conditions, the communication distance increased, but global communication was still unimaginable. The discoveries of electricity and radio-electricity in the 19<sup>th</sup> century fundamentally changed communication.

### 1.1.1 Radio communications

Radio communications have a three centuries-old history. Major scientists include James Clerk Maxwell, who described in the 1870s the modern electromagnetic theory that we still use today. Its equations explained the relationship between the *electric and magnetic fields*, using the electrical theories from Faraday, Ohm, and Ampère. Maxwell's equations [Fre16, p. 511] explain that an electric field varying in time produces a magnetic field, and *vice versa*. It theorized how an electromagnetic wave could travel through space, with an electric and a magnetic field sustaining each other. In 1887, Heinrich Hertz [Fre16, p. 28] discovered radio waves by demonstrating the effect of electromagnetic radiation through space, without any connecting wires. Later

*Radio communication finds its roots in discoveries of the 19<sup>th</sup> century.*

in the same year, Guglielmo Marconi [Fre16, p. 2] performed the first demonstration of wireless communication using radio waves.

From this first communication until today, many phenomena have been discovered, and technologies invented. In our modern world, radio communications are pervasive and essential. From television and radio broadcasting to professional communications, satellite, aircraft, and marine transmissions, our society is built on them. In addition to encoding, radio transmission leverages different modulation techniques, such as amplitude, frequency, or phase modulation, detailed in Section 2.1.

Finally, radio architectures have evolved in different manners since the beginning of their design. Today, various architectures are used depending on their applications and specialization, but the more flexible ones are the **software-defined radio (SDR)** [Col+18; Gra13]. Traditionally, a radio comprises a **radio-frequency front-end (RFFE)**, connected to an analog demodulator for analog communications or a **digital signal processing (DSP)** block for digital communications. In SDRs, the radio only contains the RFFE connected to an **analog-to-digital converter (ADC)**, which will send the digitized signal to a computer. This architecture allows implementing the demodulation and the DSP in software, with all its advantages for flexibility.

However, until now, confidentiality has not been part of our preoccupations — anyone with knowledge of the encoding or the modulation can listen to the communication. Moreover, in parallel to radio communication, another type of communication was born.

### 1.1.2 Computer communications

Independently of radio communication, the rise of computers enabled the development of the ARPANET, the pioneer packet-switched network, by the DARPA in the 70s [Com14, p. 6]. With the development of the TCP/IP protocols by Vint Cerf and Robert Kahn in the 80s, and the increasing network interconnection, the global Internet began. At first, these communications were only enabled through a cable connection, but the DARPA also experimented using networks connected by radio. Today, it is common to have radio communication in the transmission chain between two devices. While it is often found between an end device and a base station for cellular networks, these radio communications may be found in “core” networks as well. While computer network protocols were not always designed with security in mind, they started to leverage cryptography for communication protection in the 90s.

*Transitioning from the telegraph to modern mobile communication took two centuries.*

*In this thesis, we focus on Software Defined Radio, where the signal processing is implemented in software.*

*Transitioning from a local network to a global Internet took two decades.*

## 1.2 MODERN SECURE COMMUNICATIONS

*Cryptography is a cornerstone of modern secure communications.*

Radio communications for commercial applications — excluding military transmissions — have a long story about using simple scrambling or coding mechanisms for analog communications. When the secret is the scrambling algorithm itself or an hard coded “key”, it can be considered as security by obscurity.<sup>1</sup> Historically, even when the only secret have been pre-shared key, those mechanisms have tended to be simple and easily brute-forceable because of the limited space of possibilities — a low entropy. On the other side, cryptography is able to mathematically guarantee a level of security strength. Today, cryptology and all its security properties, *i.e.*, confidentiality, authenticity, and integrity are considered as the standard way to protect communications. However, secure communications are facing emerging threats that arise from unexpected interactions between the software, the hardware, and its physical environment.

### 1.2.1 Cryptography for Secure Communications

*The rise of modern cryptography is due to Hellman and Diffie in the 70s.*

Cryptography is the science of protecting information by making it unintelligible for someone who is not supposed to see it, guaranteeing confidentiality. It is based on mathematical structures, such as finite fields, and algorithms applied to digital information, to transform a cleartext into a cyphertext and *vice-versa* — at the condition of having the correct key. It finds its roots in simple operations, such as transposition used by the *Scytale*, already used during antiquity to protect sensible messages. Therefore, the cryptography literature was often published in secrecy by governments or military organizations, especially during the two World Wars. Modern cryptography can be claimed to have started during the 70s, when Hellman and Diffie proposed public-key cryptography [Sch96, Foreword]. A typical example of modern cryptography usage is to use a symmetric cryptography algorithm — such as AES — to secure the channel content, while using public cryptography to exchange keys — such as the Diffie–Hellman key exchange. While cryptography is widespread in computer communications, radio communication has been late in adopting cryptography, requiring digital instead of analog radio communications protocols.

### 1.2.2 Side-channel attacks

The side channel attack idea takes its roots in a classical scenario. Let’s imagine the following: a criminal wants to steal what is inside a protected vault. To achieve this, he can turn the lock until finding the right

<sup>1</sup> According to Kerckhoff’s principle, security by obscurity is defined by a secret algorithm, while only the key should be secret [TW11, p. 768]

position for a certain number of digits. However, there are too many possibilities to be achieved by testing all of them. Lucky he is, he pays attention to the fact that when the lock is moving on the position for one digit, the sound emitted is not the same as the one when it is not in the right position. Thanks to this information, side-channel information about the locking system, he will be able to open the vault without knowing the key beforehand.

From this imaged scenario, we can understand that a side-channel attack is an attack against a security system using additional information, originating from the interaction between the security system and its physical environment — in this case, the sound, being an acoustic side channel. This is why it is a “side channel”, and not a “main channel”, because this additional information is leaked in an unintended way.

### 1.2.2.1 Attacks Against Secret Cryptographic Keys

Cryptographic algorithms have been the first targets of side-channel attacks, using various side channels. In 1996, *Kocher et al.* [Koc96] exploited the time taken by the attacked cryptographic algorithm to infer its secret key. In 1999, they performed another attack using its power consumption [KJJ99]. In addition to time and power consumption, various classes of side-channels were discovered and exploited, e.g.: electromagnetic emanations [QS01], acoustic waves [GST17], or the optical photonic emanations [FHo8]. Moreover, various algorithms have been attacked through side-channel attacks. None of them are resistant to such attacks by design but need specific countermeasures. AES is maybe the most attacked algorithm for more than twenty years in various ways [Man03; Ors+04; LBC16; Zed22]. However, other symmetric algorithms are also attacked, such as DES [Sta10], and asymmetric algorithms, such as RSA [Rug15].

### 1.2.2.2 Attacks Against Non-Cryptographic Information

Not only cryptographic keys can be inferred from side-channel attacks, but more generally, any data with a data-dependent leakage can be inferred as well. One of these leaks, for example, is the electromagnetic radiation caused by electronic circuits. Eavesdropping through electromagnetic emanation has first been discovered by the Bell Lab during the World War II, and re-discovered during the 50s by the CIA [Age72]. In documents declassified in December 2000, the NSA uncovers its standard to offensively exploit and defensively suppress compromising electromagnetic emanations, known as *TEMPEST* [Ros82]. In 1985, *Van Eck* discovered the same phenomenon to eavesdrop on the image displayed on a computer screen, leveraging the leakage from the monitor. Published in the public domain, it is today known as *Van Eck Phreaking* [Van85]. Other non-cryptographic information that can be inferred from electromagnetic emanations, among others, are audio

*Side-channel attack exploits additional information leaked through the interaction of a system and its environment.*

*Today, the most-used cryptographic algorithms are vulnerable to side-channel without specific and costly countermeasures.*

*In addition to cryptographic algorithms, non-cryptographic information transitioning in electronic devices is also at risk.*



processed by earbuds [CYC20], keystrokes typed on wired and wireless keyboards [VP09], or data transferred on internal buses [Dan+24].

### 1.3 ON INTERDISCIPLINARITY

#### 1.3.1 Emerging Threats at the Boundaries

Most of the cited security issues have been known for a long time — around 70 years concerning compromising electromagnetic emanations. However, it does not imply that the problem is well understood. On the contrary, since a security issue may arise from unintentional interactions, it is still difficult if not impossible to predict these issues with the increasing complexity of modern electronic devices.

For example, in 2005, timing side-channel attacks were conducted by Bernstein *et al.* on AES [Ber05], exploiting the timing differences when the processor is loading a value from the memory or its cache. Cache side channels were then used recently, in 2018, in *Spectre-like* [Koc+19; Koc+20] attacks exploiting speculative execution. This kind of attack exploits transient execution after a misprediction from the branch prediction unit of the processor concerning future executed instructions. This class of vulnerability allows to bypass software-defined security barriers and is particularly complex to fix because the cause of the transient instruction is not a bug — it is precisely what the processor is supposed to do, to optimize performance.

Another recent example is the long-distance electromagnetic side-channels *Screaming Channels* [Cam+18; CFS20]. This attack exploits a coupling path between the digital and analog block inside a mixed-signal SoC, a security issue due to the miniaturization and integration of this kind of electronic device. Back in the 90s, Kuhn and Anderson discovered that a user software can control the shape of unintentional electromagnetic emanations depending on the chosen instructions, called *Soft-TEMPEST* [KA98; AK99]. This discovery leads to several applications, such as the covert channels of Guri *et al.* between 2015 and 2020 between air-gaped computers [Gur+15; GME16; Gur23]. As a further research of *Soft-TEMPEST*, Camurati and Francillon discovered a way of shaping arbitrary electromagnetic emanations in 2022, called *Noise-SDR* [CF22].

Another kind of threat emerging from the boundaries is the so-called *Out-of-Band* signal injection attacks [GR20] in the context of sensors. This type of attack allows an attacker to manipulate a sensor interface which is not intended for communication through a signal injection, to modify in a controlled way the digitized version of the measured physical quantity.

Finally, Cayre *et al.* discovered a cross-protocol attack called *WazaBee* [Cay+21b] in 2021. This attack enables communication and piv-

*Side channels on the processor microarchitecture have been rising since the Spectre and Meltdown attacks.*

*Leveraging electromagnetic emanations, long-distance side channels, and information exfiltration through cover channels are also expanding.*

otal attacks between [Bluetooth Low Energy \(BLE\)](#) and ZigBee devices, because of unexpected similarities between the two modulation techniques.

As a conclusion of this limited literature review, we observe that all of these attacks or techniques do not exploit any bugs in software or hardware. Instead of “conventional” attacks in software security, all these security issues, depicted as unexpected before their discovery, exploit in reality an expected behavior of the hardware or the physics.

### 1.3.2 Convergence of Complementary Disciplines

From the non-exhaustive examples depicted in the previous section, we understand that cross-layer issues need to be addressed using an interdisciplinary analysis.

**EXAMPLE OF A MODERN SoC** As an example, we can take a look at a modern [system-on-chip \(SoC\)](#) illustrated in Fig. 1.2. First, we observe that a typical SoC is composed of fully digital, analog, and mixed analog-digital blocks. This category is typically called *Mixed-Signal System-on-Chip (MSoC)*, and are exploited in *Screaming Channels-like* attacks [[Cam+18](#); [GKT19](#); [Dan+24](#); [CYC20](#)]. Second, considering security issues arising from the interactions between all of those blocks with each other and with their physical environment, we realized that a lot of disciplines are involved. Particularly, the *CRYPTO* block is implementing the cryptographic algorithms in hardware, while the *CPU* block may be used to implement them in software. Moreover, the *RADIO* block receives and transmits radio waves to other devices. Finally, the *POWER* block is shared between all the different blocks, which may lead to crosstalk due to coupling between different blocks, in particular through substrate coupling [[CYC20](#)]. Considering these issues systematically implies to consider concepts and theories from electromagnetism, telecommunications, electronics, computer architecture, and cryptography.

*A modern system-on-chip contains a mix of heterogeneous hardware blocks.*

## 1.4 CONTRIBUTIONS

**RESEARCH QUESTION** In the context of security issues arising from interactions between different abstraction layers, this thesis focuses on electromagnetic emanations as vectors of side-channel information. In particular, we investigate novel attacks using [software-defined radios \(SDRs\)](#) and side-channel analysis. The research question that is guiding the work of this thesis is the following:

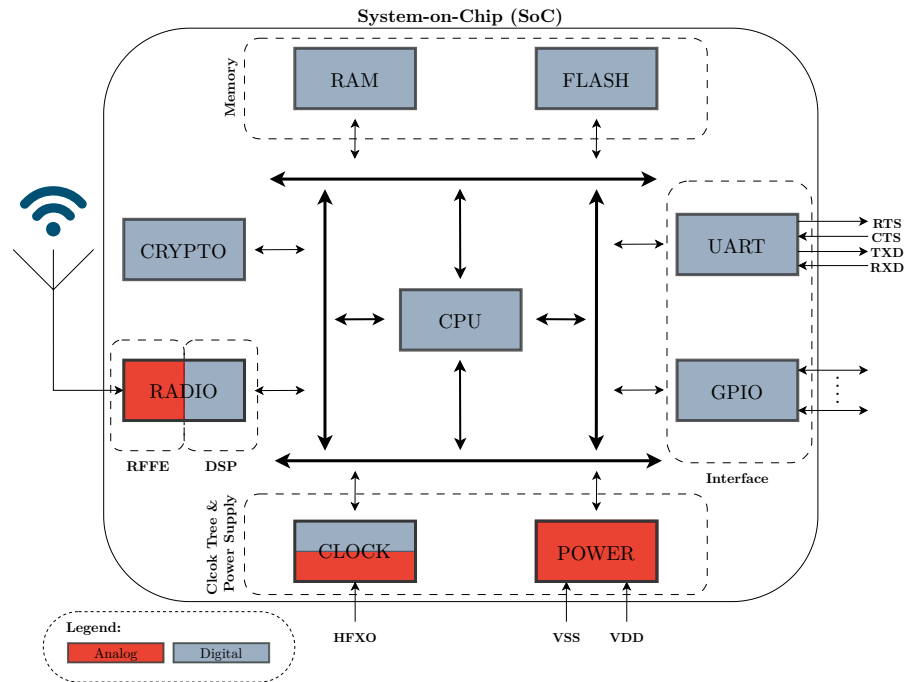


Figure 1.2: Block diagram of a Mixed-Signal System-on-Chip (MSoC).

How does digital activity modulate electromagnetic emanations resulting in information leakage and is it a threat to wireless communication protocols?

Each part will then define a more precise research question.

*Our approach is oriented towards understanding modulation leakage because of digital activity and assessing its impact in modern protocols.*

**OBJECTIVES** This research question leads to the following objectives:

- Assessing to which extent new side channel attacks that leverage electromagnetic emanations are a threat to modern **Internet of Things** protocols. Wireless communication protocols may not consider electromagnetic side channels as a realistic threat, especially when the threat model acknowledges the premise that an attacker will not be able to attack from a short distance. We tackle this question using a low-level traffic injection approach to enable the Screaming Channels effect on a modern IoT protocol.
- Discover, understand and evaluate new vectors of electromagnetic side-channel attacks. Having a better understanding of how digital activity modulates unintentional electromagnetic emanation would lead to better attacks and defenses. From the offensive perspective, it would allow to uncover new attack vectors or improve the already existing ones. From the defensive perspective, it would allow to design efficient countermeasures and leverage this understanding for preventive applications — *e.g.*, side-channel automatic detection.



- Investigate different hardware tools and create new software tools to assess those security threats. Both hardware and software tools are flourishing, however, no standard, platform, or framework is extensively used in the community. At our level, this thesis tries to build a better environment for assessing those security threats.

This thesis is organized as two academic contributions. As part of these research projects, we also performed tooling development and data collection, which we both open-sourced.

#### 1.4.1 Academic Contributions

**BLUESCREAM** The first contribution of this thesis is called *BlueScream: Screaming Channels on Bluetooth Low Energy*, which is the assessment of the *Screaming Channels* attack on the *Bluetooth Low Energy* protocol. Screaming Channels enable long-distance electromagnetic side-channel attacks by leveraging a leakage broadcasted by the radio transceiver. However, modern wireless communication protocols tend to have short transmission times, to save power and efficiently use the electromagnetic spectrum. Consequently, while this attack could be a significant threat to [Internet of Things](#) protocols, its impact remains unexplored. This work demonstrates how the protocol can be manipulated by a malicious actor through low-level traffic injection to force the victim to broadcast the leakage exploited by Screaming Channels. Leveraging this technique, we evaluated the Screaming Channels attack during an end-to-end attack using real-world firmware against the AES used for message encryption. While the attack shows performance limitations, this work warns against this threat for future wireless communication.

*This contribution enables and evaluate the long-distance side-channel Screaming Channels on the modern Bluetooth Low Energy IoT protocol.*

**PHASESCA** The second contribution of this thesis is called *PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels*, which demonstrates a novel side-channel leakage source through unintended phase modulation of electromagnetic signal. It is well-known that side-channel information leakage occurs through amplitude modulation of unintended electromagnetic emanations. However, other forms of modulations are not evaluated by the literature in the context of unintentional electromagnetic emanations. This work shows that tested electronic devices produce phase-modulated electromagnetic emanations that can be exploited in side-channel attacks. Not only this has never been demonstrated before, but building on the existing literature, we show that both amplitude-modulated and phase-modulated leakage can be combined to improve the existing attack performance. In addition, this work also theoretically explains and experimentally evaluates the relationship between the phase-modulated leakage and the jitter produced by the digital activity of the device. While the jitter has already been exploited in the state-of-the-art, we created a bridge between old and recent literature that were not aware of each other. Finally, on a more

*This contribution uncovers phase-modulated electromagnetic emanations as a new side-channel vector.*

practical aspect, electromagnetic side-channel attacks on [system-on-chip \(SoC\)](#) are traditionally performed using oscilloscopes. In this work, we use [software-defined radios \(SDRs\)](#) and theoretically compare them to oscilloscopes on how they can be used and be well-suited to assess these security issues.

### 1.4.2 Open-Source Tools & Data

Every tool developed and data collected during this thesis is open-sourced:

- Instrumenting dongles and [SDRs](#) implies having code to perform data acquisition, setup instrumentation, and side-channel attacks. The code used during this thesis is first directly included inside the research project discussed above, while a part of it has been released in separate tools.<sup>2</sup>
- Working on side-channel attacks means collecting, process, and exploiting datasets of tens of thousands of traces. In both cases, for our first contribution on Screaming Channels<sup>3</sup> and our second contribution PhaseSCA<sup>4</sup>, we open-sourced our datasets and the code used to collect and exploit them. While several terabytes of data have been collected during this thesis, our final dataset storage retains one terabyte of quality data.

Our contributions explore specific security threats, but their relevance lies in their generalization, in both offensive and defensive perspectives. First, they allow us to have a deeper understanding of how digital activity produces unintended electromagnetic emanations, which is essential to assess side-channel attacks correctly and to protect against them. Without a deep comprehension of the internal effects that produce those leakages, we would not be able to have a deep cover of those threats — as demonstrated by *PhaseSCA*, which exploits a leakage that the literature gives not enough attention because electronic devices are often exploited as black boxes. Second, they allow us to have a broad overview of the impact of electromagnetic leakage on future protocols, which is essential to build protocol specifications systematically resilient to side-channel attacks. Indeed, while electromagnetic emanations were considered as a threat only in short distance threat model, we show that this must be re-considered to have a strong security guarantee for the future.

*We released our tools and data in open-source for reproducible research sustainability.*

### 1.4.3 Outline

This thesis is organized into five parts.

<sup>2</sup> Tools listed at: [https://s3.eurecom.fr/~pierre\\_ay/](https://s3.eurecom.fr/~pierre_ay/)

<sup>3</sup> BlueScream code: [https://github.com/pierreay/screaming\\_channels\\_ble](https://github.com/pierreay/screaming_channels_ble)

<sup>4</sup> PhaseSCA code: [https://github.com/pierreay/phase\\_data](https://github.com/pierreay/phase_data)

First, Part **i** is not based on academic publications, and presents the background and the state-of-the-art of emission security. The background presents the knowledge, coming from the different disciplines mentioned above, needed to understand the state of art and our contributions. The state-of-the-art will present the history and the most recent research about electromagnetic emissions security and side channels.

The two following parts are based on two academic publications, presented at the beginning of this thesis.

PART **ii** depicts the first main contribution of this thesis, exploring the impact of the Screaming Channels attack on the Bluetooth Low Energy protocol. This part first introduces the motivations for delving into such a research project, then explains the end-to-end attack.

PART **iii** introduces the second main contribution, demonstrating an unexplored modulation being a novel source of electromagnetic side-channel. In the beginning, it explains how we discovered this new phenomenon and its relationship with Screaming Channels, before diving into the evaluation.

Last, Part **iv** presents the final reflections about those contributions. This part discusses our work, in particular its limitations, the promising future work, and our conclusion.



## Part I

### OVERVIEW OF ELECTROMAGNETIC SIDE CHANNELS

Computer security studies offensive and defensive approaches that encompass several layers of abstraction. Ultimately, an attack defeats a security system to disclose protected information, abuse a device, spoof a user identity, or other malicious goals. Physical and hardware attacks live at the boundaries between computer science and electronics. More specifically, to fully explain the phenomena incriminated in electromagnetic attacks, radio and physics knowledge has to be inevitably considered.

In this thesis, we are studying such security issues at the intersection between several disciplines. As such, in this part, we will first introduce the background knowledge needed for the reader to understand our contributions and the state of the art. Last, we will present the state of the art of attack leveraging compromising electromagnetic emanations, mainly through side-channel analysis.

*Electrons Do Not Read Schematics*

— Clayton R. Paul [[Pau06](#), p. 763]



# 2 | BACKGROUND

UNINTENTIONAL emanations understanding and exploitation leverage the knowledge of multiple disciplines that we will present in this chapter. First, Section 2.1 introduces the basics of radio communication and signal processing, allowing the reader to understand how we receive and exploit the modulation of [compromising emanation](#). Second, Section 2.2 present [electromagnetic compatibility](#), the discipline that will allow the reader to understand why and how [compromising emanation](#) are produced by embedded devices. Finally, Section 2.3 presents the [AES](#) encryption standard and how it is exploited in side-channel attacks.

## 2.1 RADIO COMMUNICATION

Radio communication encompasses the science of studying how [radio-frequency wave](#) behaves and engineering enabling the use of those waves to transmit information. Electromagnetism, [radio-frequency](#) electronic architecture and [digital signal processing \(DSP\)](#) are the disciplines that allow us to design radio systems in order to communicate. In this section, we will depict some basic radio knowledge needed to understand how a signal is acquired, represented, and processed.<sup>1</sup>

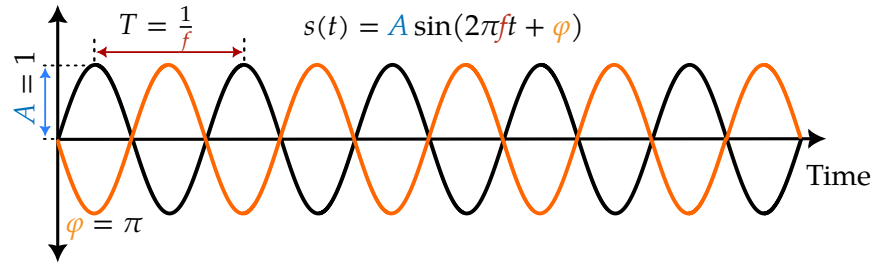
### 2.1.1 *Signal Representation*

A physical oscillating waveform is mathematically described as a *signal*. A signal can be modeled using multiple conventions, such as real values and complex numbers. They are all equivalents in representing the physical phenomenon of a waveform, but using one representation over another can be more appropriate depending on the application (*e.g.*, hardware implementation of modulator or demodulation, visualization

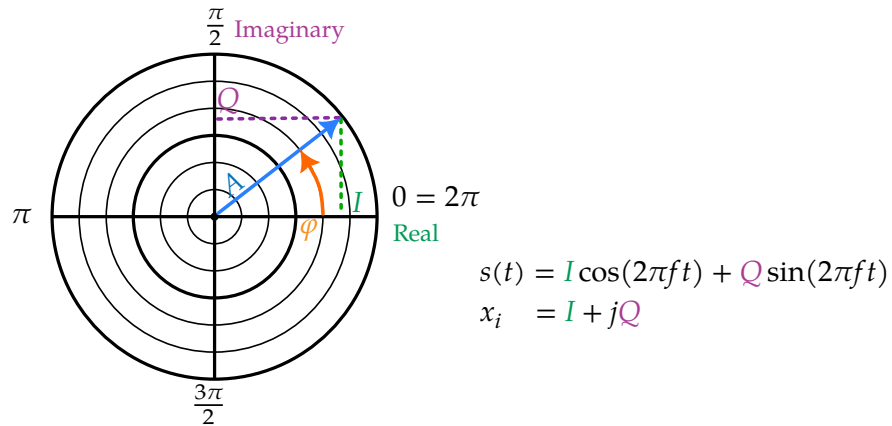
---

<sup>1</sup> As further reading about radio communication, the reader can refer to:

- Kennedy's book "Electronic Communication Systems" [Ken11],
- Frenzel's book "Principles of Electronic Communication Systems" [Fre16],
- Molisch's book "Wireless Communications" [Mol11].



**Figure 2.1:** Representation of a real-valued signal. A difference in amplitude or in period (*i.e.*, inverse of frequency) are delimited using dashes, while a difference in instantaneous phase is illustrated using a shift of  $\pi$  for the orange signal.



**Figure 2.2:** Representation of a complex-valued signal (*i.e.*, analytic representation).

of signal property on a phasor). A signal can be defined as a real-valued function of time  $s(t)$  in Equation 2.1 (illustrated by Fig. 2.1):<sup>2</sup>

$$s(t) = A \cos(2\pi ft + \varphi) \tag{2.1}$$

with:

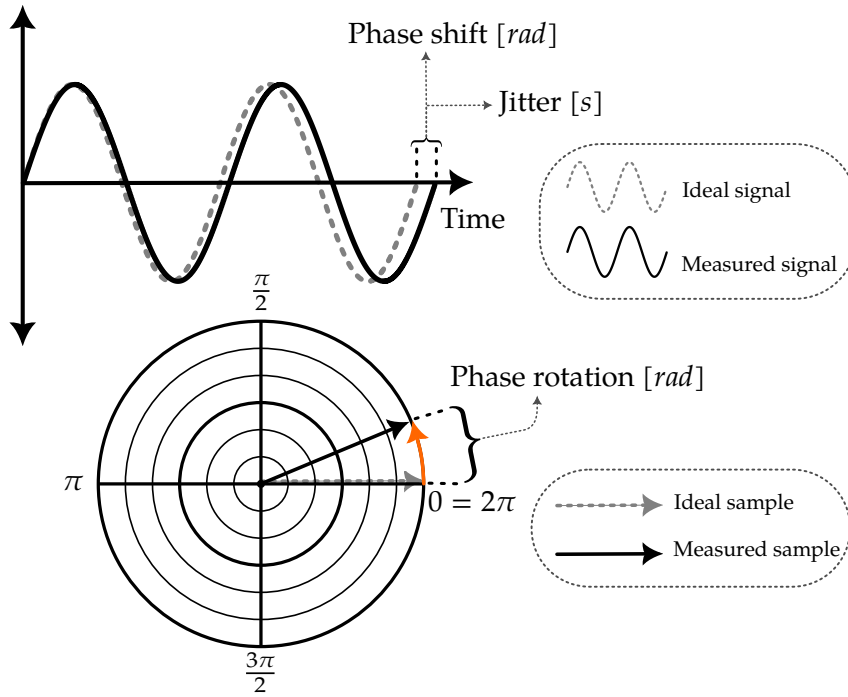
- $A \in [0, +\infty]$  Amplitude,
- $f \in [0, +\infty]$  Hz Frequency,
- $\varphi \in [-\pi, \pi]$  rad Phase.

The amplitude is representing the magnitude of the variations of the periodic function in a single period. The frequency is representing the temporal rate of change of the instantaneous phase, *i.e.*, the time derivative of phase. The phase is representing the angle quantity representing the fraction of the cycle covered up.

<sup>2</sup> For further reading about digital signal processing, the reader may refer to:

- Richard Lyons' book "Understanding Digital Signal Processing" [Lyo10],
- Smith's book "The Scientist and Engineer's Guide to Digital Signal Processing" [Smi99].





**Figure 2.3:** Depending on the expression unit (which depends on the measurement tool), a difference in two phase values will be expressed or represented in different ways.

Because of trigonometric identities, the same signal can be defined as the sum of a *sine* and a *cosine* functions with both a phase of 0, as shown in Equation 2.2 [Lyo08; Put18] [Col+18, p. 37] (illustrated by Fig. 2.2):

$$s(t) = I \cos(2\pi ft) + Q \sin(2\pi ft) \quad (2.2)$$

With  $I \in \mathbb{R}$  and  $Q \in \mathbb{R}$  representing the amplitudes of the two signals. Considering a signal recorded using **in-phase and quadrature (I/Q)** sampling, it will be sampled using the analytic representation. Therefore, our discrete signal is a complex-valued function  $x(t)$  where each sample  $x_i$  is a complex number of the form  $x_i = I + jQ$ , with  $I$  the real part,  $Q$  the imaginary part and  $j$  the imaginary unit. With a discrete complex-valued signal, the amplitude can be computed as the quadratic sum of the  $I$  and  $Q$  samples, as shown in Equation 2.3:

$$A = \sqrt{I^2 + Q^2} \quad (2.3)$$

**PHASE AND JITTER** As illustrated in Fig. 2.3, the difference between two phase values may be designated using *phase shift* when represented in the 1D time-domain or *phasor rotation* when being represented in the 2D complex plane. When measuring in radian, the difference between two phase values is expressed as *phase shift*, while it is expressed as *jitter* when measured in seconds.

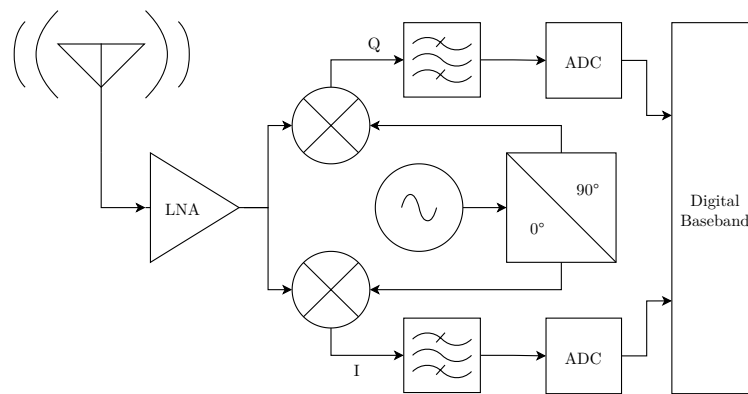


Figure 2.4: Radio receiver direct-conversion architecture.

### 2.1.2 Radio Architecture

The objective of a radio receiver is to sample a signal at **baseband** frequency (lower frequency) from a carrier frequency (higher frequency). Figure 2.4 shows a typical architecture of a direct-conversion radio receiver [Beh07]. A received signal, *via* an antenna or a probe, is amplified through a **LNA**. It is then mixed with two in-quadrature signals, explained in Section 2.1.1, to perform the down-conversion, *i.e.*, lowering the frequency from the carrier to the **baseband**. In-quadrature corresponds to two signals with a phase shift of  $\frac{\pi}{2}$  (or  $90^\circ$ ), hence for sinusoidal functions, it corresponds to one *sine* and one *cosine* functions respectively. The resulting I and Q signals are then filtered and sampled by an **analog-to-digital converter (ADC)**, which results in samples that are stored as complex numbers – where the I is the real part and the Q is the imaginary part.<sup>3</sup>

### 2.1.3 Measurement Equipment for Side Channels

While differing in architecture, oscilloscopes and radios (introduced in Section 2.1.2) are common measuring instruments for **electromagnetic (EM)** side channels.

**OSCILLOSCOPE** The most used equipment for performing side channels are oscilloscopes [HH15, p. 1158]. It is an equipment able to sample a large bandwidth of frequencies with a duration dependent on the

<sup>3</sup> For further reading about radio architectures and engineering, the reader may refer to:

- Razavi's book "RF Microelectronics" [Raz12],
- Nguyen's book "Radio-Frequency Integrated-Circuit Engineering" [Ngu15],
- Behzad's book "Wireless LAN Radios" [Beh07],
- Collins' book "Software-Defined Radio for Engineers" [Col+18],
- Grayver's book "Implementing Software Defined Radio" [Gra13].

sample rate and its buffer size. The sampling bandwidth ranges from 0 to  $\frac{sr}{2}$  with  $sr$  being the sampling rate according to the Nyquist–Shannon sampling theorem. Sampling a large bandwidth is achieved by using an **ADC** with a very high sample rate – *e.g.*, several GHz, implying that the **ADC** is a costly circuit. In the side channels context, the acquisition of a trace is generally initiated using an accurate digital trigger. The oscilloscope generates a 1D real-valued vector corresponding to the sampled signal (using the real-valued representation defined in Section 2.1.1). This trace can then be used for a side-channel attack.

**RADIO** Performing a side-channel attack using a radio equipment, such as a **SDR**, involves two standard steps:

1. Acquire the signal through sampling and measurement with an EM probe, resulting in a complex-valued vector – the **in-phase and quadrature (I/Q)** analytic representation as explained in Section 2.1.1.
2. Compute the amplitude of the complex signal, resulting in a 1D real-valued vector, called a “trace” when represented in time-domain – see Equation 2.3. In telecommunication terms, this is analogous to performing amplitude demodulation, using each amplitude value as a symbol. This trace can then be used for a side-channel attack.

**COMPARISON** In contrast to a radio architecture exposed in Section 2.1.2, an oscilloscope does not perform any down-conversion process from an intermediate frequency to the **baseband**. As a result, to reach the same frequency, the local oscillator and mixer need to be replaced by a costly **ADC**. Compared to an oscilloscope, suited to acquire a signal on a very large band, a **SDR** can acquire a signal on a narrow band with a better sensitivity. Considering the accurate digital trigger, the absence of the down-conversion process, and the 1D real-valued sampled vector, the oscilloscope output signal can be directly used for a side-channel attack compared to the **SDR** output signal, which has to be pre-processed.

## 2.2 ELECTROMAGNETIC COMPATIBILITY (EMC)

Electromagnetic compatibility [IEC18b] is a field of electronic engineering that studies the **electromagnetic radiation (EMR)** of a **device under test (DUT)** through compliance tests — such as the FCC.<sup>4</sup> These tests

<sup>4</sup> As further reading about **electromagnetic compatibility (EMC)**, the reader is encouraged to read:

- “Introduction to Electromagnetic Compatibility” from Clayton R. Paul [Pau06],
- “High-Speed Digital Design: A Handbook of Black Magic” from Johnson and Graham [JG93],

ensure that the [electromagnetic radiation](#) do not exceed certain limits (for safety or health reasons), such that the [DUT](#) will not cause disturbances of nearby devices or be harmful to health. They also ensure that the [DUT](#) will be able to work in challenging [radio-frequency](#) environment, even despite [intentional electromagnetic interference \(IEMI\)](#). In summary, [electromagnetic compatibility \(EMC\)](#) ensures limiting [electromagnetic radiation](#) generation and propagation to limit [electromagnetic interference](#) from a functional point of view through various techniques (introduced in Section 10.1.1 from Part iv). The literature of [electromagnetic](#) attack often employs [EMC](#) terminology using approximated meanings, even though the International Electrotechnical Commission (IEC) is standardizing the terminology. In this section, we will strive to carefully define [EMC](#) phenomena that are relevant to [compromising emanation](#).

### 2.2.1 Interference

[Electromagnetic compatibility](#) is about limiting [electromagnetic \(EM\) interferences](#) caused by [EM disturbance](#). As defined by the IEC [[IEC18d](#)], an interference is a degradation in performance because of a disturbance. More generally, some documents define interference as the effect of a disturbance in one circuit creating undesired variations in another circuit. As defined by the IEC [[IEC18c](#)], a disturbance may be a *noise* or an *unwanted signal*. Therefore, the disturbance is the cause while the interference is the consequence seen from the receiver point of view — and should not be used indiscriminately, as stated by the IEC [[IEC18c](#)]. Numerous phenomena can cause interferences and generate disturbances. Consequently, in this thesis, we will only present the main causes that are identified by the [EM](#) security community:

- Unintentional [coupling](#), introduced in Section 2.2.5,
- Unwanted non-linearities, introduced in Section 2.2.6.

To do so, we first need to introduce basic concepts of electromagnetism.<sup>5</sup>

### 2.2.2 Emanation

In its most general physical form, an emanation can be defined as energy measured in Joule ( $J$ ) transferred from one system to another. While

- “Electromagnetics Explained” from Schmitt [[Scho2](#), ch.12].

<sup>5</sup> For a deeper introduction to electromagnetism and Maxwell equations, the reader may refer to:

- “Electromagnetics Explained” book from Schmitt [[Scho2](#)],
- “Fundamentals of Physics” book from Jearl Walker [[Wal14](#)],
- “Fundamentals of Electromagnetics” book chapter from Nguyen [[Ngu15](#)].

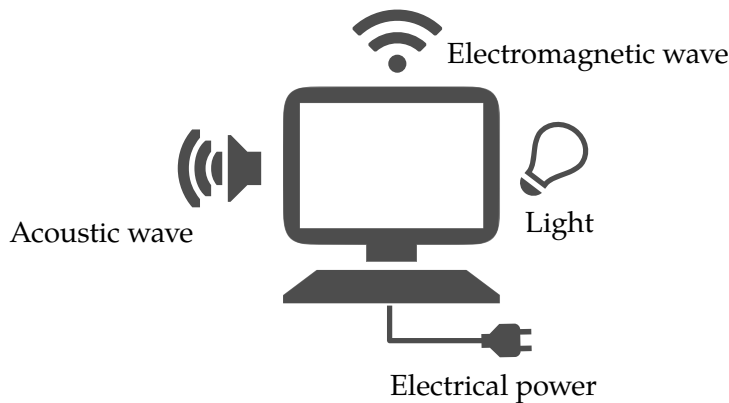


Figure 2.5: Different forms and media where an unintentional emanations may propagate through.

emanations also encompass energy transferred through mechanical work as acoustic waves (as shown in Fig. 2.5), in this thesis, we are focusing on energy transfer in the form of **electromagnetic radiation (EMR)**. As defined by IEC, an **electromagnetic radiation [IEC90]** is an energy transfer in the form of **electromagnetic wave** traveling through space, described by the Maxwell equations. However, as stated by IEC, it sometimes encompasses induction phenomenon — when the energy is stored locally around a component acting like a capacitor or an inductor. In this thesis, we employ the term **electromagnetic radiation** for the two forms of energy transfers, while trying to always specify which type of emanations is concerned.

In the following sections, we will see how the energy of an emanation is transferred:

- Through the electric or magnetic field, described in Section 2.2.3.
- Depending on the spatial region, also known as *near-field* and *far-field*, described in Section 2.2.4.
- Depending on the transfer mode, also known as **coupling**, described in Section 2.2.5.

### 2.2.3 Electric Field and Magnetic Field

A field is a physical quantity represented by a scalar (such as temperature or pressure) or a vector, depending on if the measured physical property possess a direction. Electric and magnetic fields are mathematically defined as a *vector quantity*, because they convey both the information of a *force*, *i.e.*, its *magnitude* and its *direction*. Electric and magnetic fields are usually represented using a bold type, which is equivalent to the vector notation. It is important to have a basic understand of those two

fields to understand how energy is transferred between systems, and finally, how compromising leakage happens.

**ELECTRIC FIELD  $\mathbf{E}$**  Any charged particle will cause an electric field around it, through which the particle will apply an *electrostatic force* ( $\vec{F}$ ) to another charged particle in that field. The electric field can be defined through the electrostatic force applied on a test charged particle, defined in Eq. (2.4) [Wal14, ch.22]:

**Definition 2.2.1** (Electric field vector).

$$\mathbf{E} = \frac{\vec{F}}{q_0} \quad (2.4)$$

with:

- $E$  Electric field vector,
- $q_0$  Test charge,
- $\vec{F}$  Electrostatic force applied on  $q_0$ .

The directions of the electric field lines originate from positively charged particles and terminate on negatively charged particles. The magnitude of the electric field set up by a particle with charge  $q$  expressed in Coulomb (C) at a distance  $r$  from the particle is given by Eq. (2.5) [Wal14, p. 633]:

**Definition 2.2.2** (Electric field magnitude).

$$E = \frac{1}{4\pi\epsilon_0} \frac{|q|}{r^2} \quad (2.5)$$

with:

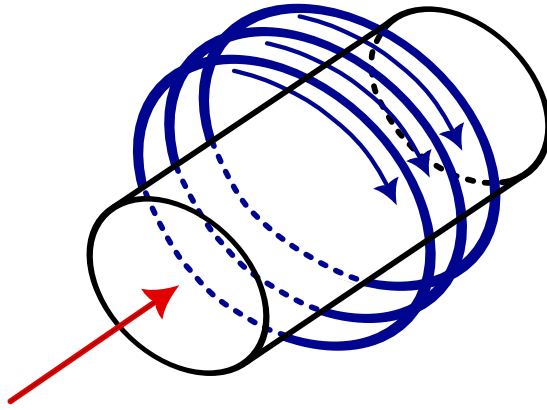
- $E$  Electric field magnitude,
- $\epsilon_0$  Permittivity constant,
- $|q|$  Positive charged particle,
- $r$  Distance to the charged particle.

Therefore, the magnitude of the electric field is proportional to the value of the charge and follows an inverse-square law<sup>6</sup>.

**CURRENT** The electric current  $i$  is defined by the rate (over time) at which charge ( $q$ ) passes a point in space, defined in Eq. (2.6) [Wal14, p. 614, p. 622]:

$$i = \frac{dq}{dt} \quad (2.6)$$

<sup>6</sup> An inverse-square law defines any physical quantity inversely proportional to the square of the distance from the source of the physical quantity.



**Figure 2.6:** A current (red) flowing across a conductor will induce a magnetic flux (blue) in its vicinity.

**MAGNETIC FIELD ( $\mathbf{B}$ )** On the contrary of electrically charged particles, there is no “magnetically charged particle” — to the knowledge of physics at the first quarter of the 21<sup>st</sup> century. That being said, there is two methods to produce a magnetic field [Wal14, p.804]:

- *Electromagnet:* A magnetic field is caused by moving charged particles.
- *Permanent magnet:* A magnetic field is caused by an object in which the *intrinsic* magnetic fields of particles are not canceling out (forming a magnetic dipole).

The one that interests us is the *magnetic field due to current* [Wal14, ch. 29], as illustrated in Fig. 2.6. Since a moving charged particle produces a magnetic field around itself, a current of moving charged particles produces a magnetic field around the current. The *Biot–Savart law* describe the magnetic field magnitude (in its scalar form, defined in Eq. (2.7)) and direction (in its vectorial form) for a given arbitrary current [Wal14, p. 837] at a particular point:

**Definition 2.2.3** (Magnetic field magnitude (Biot–Savart law)).

$$dB = \frac{\mu_0}{4\pi} \frac{i ds \sin \theta}{r^2} \quad (2.7)$$

with:

- $B$  Magnetic field magnitude,
- $\mu_0$  Permeability constant,
- $i$  Electric current,
- $ds$  Magnitude of  $\vec{ds}$  current-length vector (portion carrying  $i$ ),
- $r$  Magnitude of  $\vec{r}$  pointing toward  $\mathbf{B}$  from  $ds$ ,
- $\theta$  Angle between direction of  $\vec{ds}$  and  $r$ .

It is interesting to note that, for a current point towards ( $\theta = 0$ ) or away ( $\theta = 180$ ) the point of measurement, the resulting  $\mathbf{B}$  field is null since  $\sin(0) = \sin(180) = 0$ . Therefore, magnetic field lines describe circles around the current and have no start or end. Moreover, with observed that for a fixed point around a fixed wire, the only variations in the magnetic field are due to time-variations of the current  $i(t)$ .

Now that we have a basic understanding of how electric and magnetic fields are generated, we will see how they propagate depending on the considered region.

#### 2.2.4 Near-Field and Far-Field

The terms *near-field (NF)* and *far-field (FF)* describe two spatial regions in which the [electromagnetic radiation](#) will behave differently [[Fre16](#), p. 509], depending on the frequency of the [electromagnetic wave](#) and the distance from the emitter:

- *Near-field*: The region directly around the antenna. In this region, the electric field ( $\mathbf{E}$ ) and the magnetic field ( $\mathbf{B}$ ) are distinct from each other and only one is predominant over the other.
- *Far-field*: The region after the *near-field*, in which the electric field and the magnetic field are perpendicular and sustain each other. Only waves traveling in the *far-field* may be called [electromagnetic wave](#) or [radio-frequency wave](#).

The distance separating the two fields is known as the *Fraunhofer distance*.

**FRAUNHOFER-RAYLEIGH DISTANCE** The *Fraunhofer distance* (also known as *Rayleigh distance*) defines the distance at which an [electromagnetic radiation](#) is a plane wave radiating through space. As [electromagnetic wave](#) is spherical or curved by nature, it depends on the length of the antenna because it will appear as a flat wave for a small and/or distant enough antenna [[Scho2](#), p. 123].<sup>7</sup> Both the [radio transmission](#) and the [radio reception](#) antennas have to be separated by at least the Fraunhofer

<sup>7</sup> As an analogy about the relation between the dimension of an antenna and the curved nature of an electromagnetic wave, we can imagine how a basketball's surface would appear curved for a human or flat for an insect, respectively.



distance to consider a purely **far-field** transmission [Mol11, p. 48]. The Fraunhofer distance  $d_R$  is defined in Eq. (2.8) [Mol11, p.48]:

$$d_R = \frac{2D^2}{\lambda} \quad (2.8)$$

with:

- $D$  Largest antenna dimension,
- $\lambda$  Wavelength.

**FIELDS DENOMINATIONS** **Near-field** and **far-field** are sometimes called *storage field* and *radiation field* [Scho2, p. 89], respectively. The reason lies in that, for the **near-field**, a capacitor and an inductor are storing energy through an electric or magnetic field, respectively. The field will extinguish when the source power is turned off, and the shape of the field is dependent on the source circuit. In addition to *storage field*, the **near-field** is also sometimes called the *reactive field* [Scho2, p. 107]. However, in the **far-field**, the energy is radiated through space, and hence, transferred to a receiver or traveled until being completely absorbed. The field does not depend on the source power state, and the shape of the field is always spherical — but can be equivalent to a plane, as seen with the *Fraunhofer distance*. Another denominations is the *Fresnel zone* for the **near-field** and the *Fraunhofer zone* for the **far-field**.<sup>8</sup>

### 2.2.5 Coupling

A **coupling** between two systems is the phenomenon by which transmission of energy from a source system (emitter, culprit) to a sink system (receiver, victim) occurs. The coupling can be intentional, for example:

- Between two antennas in a communication system to transfer information leveraging radiating **electromagnetic wave** through the **far-field**,
- Between the stator (fixed electromagnet or permanent magnet) and the rotor (rotating coil) in a dynamo, an electrical generator from mechanical work. The stator generates a constant magnetic field, and a current will be *induced* in the rotor because of an *electromotive force*, described by the *Faraday's law of induction* [Wal14, ch. 30].

When exploiting **compromising emanations**, the attacker tries to maximize the intentional coupling between its measurement probe or its antenna and the victim system emitting the **compromising emanations**.

<sup>8</sup> This is not exactly true from a physicist point of view. In fact, there are 4 fields (or zones) [Scho2, p.107], and the Fresnel zone encompasses the first 3 zones, starting from the emitting antenna, and the Fraunhofer zone corresponds to the last one.

Moreover, inside a system, unintentional coupling occurs. Its consequences are called [electromagnetic interference](#) when generating a disturbance or [compromising emanation](#) when leaking secret information. Coupling can happen through various models and media, and we will depict them to have a basic understanding of the phenomenon that are at the root causes of [compromising emanation](#).<sup>9</sup>

### 2.2.5.1 Models

The *coupling model* corresponds to the physical mode of energy transmission, also called *coupling path*. The EMC literature classify the models as three categories: *conducted, induced, radiated* [Scho2, p. 255] [Devo8a, p 3].

**CONDUCTED** Conducted coupling is also called *direct, conductive, galvanic, or resistive* coupling depending on the authors. This coupling occurs when there is a direct electrical contact (through a conductor) between the two systems, and is modeled as a resistor connection. In electronic circuits, it often happens through voltage drops on power-supply lines or [ground](#) lines shared between two subsystems. Indeed, when considering high frequencies, a current flowing through a [ground](#) line of a non-zero impedance will cause voltage drops across the line, leading to points in its surface at different potentials. When occurring, this conducted coupling is called *common-impedance* coupling [Pau06, p. 768]. Let us consider two conductors, a source conductor called *S* and powering the *L1* load, and a victim conductor called *V* and powering the *L2* load, both sharing the ground line GND. The voltage drop across an impedance of the [ground](#) line is then defined by Eq. (2.9) [GSH20, p. 83]:

$$U_{\text{GND}} = Z_{\text{GND}} I_S = \frac{Z_{\text{GND}}}{Z_S + Z_{L1}} U_S \quad (2.9)$$

with:

$U_{\text{GND}}$	Voltage drop across ground,
$Z_{\text{GND}}$	Impedance of ground,
$I_S$	Current of source conductor,
$Z_S$	Impedance of source conductor,
$Z_{L1}$	Impedance of the load of the source conductor,
$U_S$	Voltage of electrical source.

Since the voltage drop  $U_{\text{GND}}$  happens on the ground line shared with *V*, this directly contributes to the voltage of the victim conductor *V*

<sup>9</sup> As further reading about coupling models and paths, the reader can refer to the “High-Power Electromagnetic Effects on Electronic Systems” book from Giri *et al.* [GSH20, ch. 3].

which now depends on the source conductor  $S$ . More particularly, it also depends on the load  $L1$ , which could be a digital hardware producing a switching activity — therefore, the digital activity signal would be coupled to the  $V$  conductor, hence, on the  $L2$  load.

**INDUCED** Induced coupling occurs through the [near-field](#), when there are two partially parallel conductors, typically spaced by less than a wavelength ( $\lambda$ ). The induced current or voltage is proportional to the speed variation of the field, *i.e.*, to the time-derivative of the field [[Scho2](#), p. 256].<sup>10</sup> In the following, we will denote as  $\omega = 2\pi f$  the angular frequency of the sinusoidal voltage source. Induced coupling can be described by two models [[HH15](#), p. 581]:

- *Capacitive*: Capacitive coupling occurs when a victim conductor is in the vicinity of a time-varying electrical field ( $\mathbf{E}$ ) generated by another conductor. This coupling is modeled by a capacitor between the source and the victim — *i.e.*, a mutual capacitance. The induced current on a victim conductor  $V$  by a source conductor  $S$  is defined as Eq. (2.10) [[GSH20](#), p. 84] [[Scho2](#), p. 257]:<sup>11</sup>

$$I_V = \omega C_M U_S \quad (2.10)$$

with:

- $I_V$  Current induced on victim conductor,
- $\omega$  Angular frequency of  $U_S$ ,
- $C_M$  Mutual capacitance between source and victim,
- $U_S$  Voltage of source conductor.

A detailed analysis of capacitive coupling in [CMOS integrated circuits](#) has been published by Sicard *et al.* [[SR92](#)] in 1992.

- *Inductive*: Inductive coupling is also called *magnetic* coupling. This coupling occurs when a victim conductor is in the vicinity of a time-varying magnetic field ( $\mathbf{B}$ ) generated by another conductor (due to Ampère’s circuital law). This coupling is modeled by an inductor (*e.g.*, a coil) between the source and the victim — *i.e.*, a mutual inductance. The induced voltage (due to Faraday’s law of induction) on a victim conductor  $V$  by a source conductor  $S$  is defined as Eq. (2.11) [[GSH20](#), p. 85] [[Scho2](#), p. 258]:<sup>12</sup>

$$U_V = \omega L_M I_S \quad (2.11)$$

<sup>10</sup> In other words, induced coupling is more likely to occur in higher speed systems and smaller [integrated circuits](#), which corresponds to the trend of microelectronics until 2024.

<sup>11</sup> The given equation corresponds to the first-order approximation of the partial differential form.

<sup>12</sup> See Footnote 11.

with:

- $U_V$  Voltage induced on victim conductor,
- $\omega$  Angular frequency of  $U_S$ ,
- $L_M$  Mutual inductance between source and victim,
- $I_S$  Current of source conductor.

**RADIATED** Radiated coupling is also called *radiative* coupling. This coupling occurs when there is a time-varying current in a (potentially unintentional) antenna, emitting **electromagnetic waves** (perpendicular **E** and **B** fields) that propagates in the **far-field** [HH15, p. 582] [Pau06, p. 503]. If the frequency of the **electromagnetic wave** is belonging to the range from 3 kHz to 300 GHz, it is therefore analogous to a **radio-frequency wave**. A main cause of **electromagnetic radiation** is unintentional antennas, such as wires, PCB tracks, copper cooling pipes, or other conductive structures that can act as an antenna [Pau06, p. 504]. This phenomenon is amplified by an impedance mismatch due to either a non-ideal connector or a cable bending [Ram+22, p. 2497, fig. 7]. When considering an **electromagnetic radiation**, the power received into a matched load through an antenna follows the Friss formula [GSH20, p. 73] as defined in Eq. (2.12):

**Definition 2.2.4** (Friss formula).

$$P_R = \lambda^2 \frac{G_S * G_R}{4\pi r_R} P_S \quad (2.12)$$

with:

- $P_R$  Power at receding antenna,
- $P_S$  Power of transmitting antenna.
- $\lambda$  Wavelength of incoming wave,
- $G_S$  Power gain of transmitting antenna,
- $G_R$  Power gain of receiving antenna,
- $r_R$  Distance from the transmitting antenna.

In other words, the received power is equal to the product of the antenna's effective receiving area and the power density of the incident wave.

#### 2.2.5.2 Medium

The *coupling medium* corresponds to the physical medium through which the energy transfer occurs. In the following, we present the main media that are involved in coupling phenomenon:

- *Transmission line*: A transmission line is defined as a pair conductors [Pau06, ch. 4, p. 177]. Examples of coupling across transmission lines include conducted coupling through ground plane or power-supply lines.
- *Free space*: The free space is distinguished by the [near-field](#) and the [far-field](#). In the [near-field](#), the induced coupling can happen because of varying fields. In the [far-field](#), the radiated coupling can happen through intentional or unintentional antennas.
- *Substrate*: The silicon substrate of a chip may be the victim of unintentional coupling.

**SUBSTRATE COUPLING** High-frequency but small dimensions systems, such as modern [integrated circuits](#), are concerned with phenomena that basic circuits do not have to deal with. In such systems, conducted and induced [coupling](#) effects through the silicon substrate arise because of its finite resistivity, despite the dielectric layers of the substrate. This problem is more and more important with miniaturization and mixed circuits, containing both analog and digital subsystems.<sup>13</sup>

**DIGITAL SIGNAL PROCESSING** Note that, from a signal processing perspective, in the frequency domain, coupling between two signals of frequencies  $f_1$  and  $f_2$  is equivalent to the algebraic sum  $f_1 + f_2$  [LMM05, p. 8]. However, an adequate description of the resulting signal in the time-domain will depend on the transfer function of one of the coupling models presented above.

**CONCLUDING** We have seen the main coupling models and media. In complex embedded systems, such as the ones we are studying in this thesis, [compromising emanation](#) often emanates after a combination of multiple coupling models and media. For example, a conducted coupling can first occur between the digital system and its power line, and then the power line will radiate into the [far-field](#).

---

<sup>13</sup> For a deeper analysis of substrate coupling, the reader may refer to:

- The thesis “Substrate Noise Coupling in Mixed-Signal Integrated Circuits: Compact Modeling and Grounding Strategies” from Kristiansson [Kri07],
- The paper “Substrate Coupling Noise: Modeling and Mitigation Techniques” from Parihar *et al.* [Par09],
- The book “Analysis and Solutions for Switching Noise Coupling in Mixed-Signal ICs” from Aragonès *et al.* [Xav99],
- The technical report “Substrate Coupling in RF Analog/Mixed Signal IC Design: A Review” from Vora *et al.* [Voro3].

### 2.2.6 Non-linearities

A system is linear if its output  $F(x)$  satisfy the superposition principle [Raz12, p. 9], as defined in Eq. (2.13):

**Definition 2.2.5** (Superposition principle).

$$F(ax_1 + bx_2) = aF(x_1) + bF(x_2) \quad (2.13)$$

In other words, a system is linear if its output can be described as a linear combination (or superposition) of its output (or response) to individual inputs. When an ideal electronic circuit is designed to have a linear output (*e.g.*, an amplifier), the real circuit always introduces noise and distortion to some extent [RP03, p. 9]. Undesired non-linearities will increase the non-linear behavior of electronic circuits, *e.g.*, an incorrect bias voltage that can drive an amplifier into clipping [Fre16, p. 320]. Therefore, a circuit in which distortions are at least partially due to non-linearities is a so-called non-linear circuit [IEC02]. Let us consider an input  $x(t)$  to the non-linear system  $y(t)$ , its output can be expressed by a polynomial as in Eq. (2.14) [Raz12, p. 12] [RP03, p. 23]:

**Definition 2.2.6** (Non-linear system).

$$y(x) = \alpha_0 + \alpha_1 x(t) + \alpha_2 x^2(t) + \alpha_3 x^3(t) + \dots \quad (2.14)$$

Where  $\alpha_j$  are time-invariant coefficients. Additionally to a non-linear component, non-linearities may also designate a non-linear medium on a signal path. The four main consequences of non-linearities in electronic components that interest us are:

- *Unintentional mixing*, presented in Section 2.2.6.1,
- *Harmonic distortion*, presented in Section 2.2.6.2,
- *Intermodulation distortion*, presented in Section 2.2.6.3,
- *Crossmodulation*, presented Section 2.2.6.4.

#### 2.2.6.1 Mixing

The purpose of a mixer is to perform a frequency translation (also called a shift in frequency domain), *i.e.*, convert a signal from one frequency to another [Raz12, ch. 6] [RP03, ch. 7] [Devo8b]. From a digital signal processing perspective, it is achieved by multiplying the two signals. When considering a mixer system with only one variable input (thus, a fixed second frequency), the system can be described as a linear system. However, when considering a mixer system with two variable inputs (*e.g.*, in a radio transceiver), the system can be described only as a non-linear system. Therefore, mixers are often described as “non-linear”

circuits.<sup>14</sup> In the frequency-domain, an ideal mixer  $M$  is operating as defined in Eq. (2.15):

**Definition 2.2.7** (Mixing (Frequency-domain)).

$$M(f_1, f_2) = f_1 * f_2 = (f_1 + f_2, f_1 - f_2) \quad (2.15)$$

In the time-domain, an ideal mixer  $M$  with input signals  $x_1(t)$  and  $x_2(t)$  (angular frequencies of  $\omega_1$  and  $\omega_2$ , respectively) is operating as defined in Eq. (2.16) [HH15, p. 395]:

**Definition 2.2.8** (Mixing (Time-domain)).

$$\begin{aligned} M(x_1(t), x_2(t)) &= \cos(\omega_1 t) * \cos(\omega_2 t) \\ &= \frac{1}{2} (\cos[(\omega_1 - \omega_2)t] + \cos[(\omega_1 + \omega_2)t]) \end{aligned} \quad (2.16)$$

While a mixer can be a circuit designed for this purpose, another component not designed for this purpose can partially perform a mixing operation in addition to its original operation because of non-linearities — called unintentional mixing.

### 2.2.6.2 Harmonic Distortion

When the input of a non-linear component is a signal composed of a single frequency component  $f_1$ , the output may be composed of multiples of the input frequency  $2f_1, 3f_1, \dots$  of that signal [Mar11, p. 145], so-called *harmonics*. More specifically, considering an input sinusoid  $x(t) = A \cos(\omega t)$  to the non-linear system  $y(t)$  in time-domain [Raz12, p. 14] as defined in Eq. (2.17):

$$\begin{aligned} y(t) &= \alpha_1 A \cos(\omega t) + \alpha_2 A^2 \cos^2(\omega t) + \alpha_3 A^3 \cos^3(\omega t) + \dots \\ &= \underbrace{\frac{\alpha_2 A^2}{2}}_{\text{DC}} + \underbrace{\left( \alpha_1 A + \frac{3\alpha_3 A^3}{4} \right)}_{\text{Fundamental}} \cos(\omega t) + \underbrace{\frac{\alpha_2 A^2}{2} \cos(2\omega t)}_{2^{\text{nd}} \text{ harmonic}} + \\ &\quad \underbrace{\frac{\alpha_3 A^3}{4} \cos(3\omega t)}_{3^{\text{rd}} \text{ harmonic}} + \dots \end{aligned} \quad (2.17)$$

The resulting terms are known as the following:

- *Direct Current (DC)*: Quantity of a null frequency arising from second-order component.
- *Fundamental*: Desired input frequency.
- *Harmonics*: Multiples of the input frequency.

<sup>14</sup> “Mixer are non-linear circuits” is not totally true. In fact, only a *linear time-invariant (LTI)* system cannot generate new frequency components at its output. Therefore, a mixer may be a non-linear or a time-variant system [Raz12, p. 11].

### 2.2.6.3 Intermodulation Distortion

Intermodulation is a phenomenon caused by *non-linearities* in a circuit [IEC17] [Raz12, p. 21] [Fre16, p. 320]. It arises when the input signal is composed of at least two frequency components, *i.e.*,  $f_1, f_2, \dots, f_n$ . Its output will be a signal composed of at least the fundamental frequencies of the input, in addition to *linear combinations* of the different frequency components in the input, *i.e.*,  $x * f_1 \pm y * f_2 \pm \dots \pm z * f_n$  with  $x, y, z \in \mathbb{N}$ . More precisely in the time-domain, for an input signal  $x(t) = A_1 \cos(\omega_1 t) + A_2 \cos(\omega_2 t)$ , the result of the intermodulation  $y(t)$  is defined in Eq. (2.18) [Mar11, p. 145]:

**Definition 2.2.9** (Intermodulation).

$$y(t) = \sum_{m,n \in \mathbb{Z}} A_{m,n} \cos(m\omega_1 + n\omega_2) \quad (2.18)$$

The resulting new frequencies are called *intermodulation products*, and the consequence of an intermodulation is called **intermodulation distortion (IMD)**.<sup>15</sup> The *order* of an intermodulation product is defined as the sum of the linear combination coefficients required to obtain the frequency of the intermodulation product, *i.e.*,  $m + n$ . The attentive reader may have spotted the relationship between intermodulation and mixing: a mixer output corresponds to the positive and negative second-order intermodulation products (with  $m = 1$  and  $n \pm 1$ ).

**PASSIVE INTERMODULATION** **Passive intermodulation (PIM)** corresponds to an intermodulation distortion due to non-linearities present in transmission lines, antennas, connectors, and interfaces [Fre16, p. 925]. This phenomenon is well known to happen when rusty metallic conductors are present on the signal path, hence also known as the *Rusty Bolt Effect* [SF].

### 2.2.6.4 Crossmodulation

Crossmodulation is a phenomenon caused by *non-linearities* in a circuit and arises when the input signal is composed of at least two frequency components, just as intermodulation (presented in Section 2.2.6.3).<sup>16</sup> If one of the frequency component is modulated by a **baseband** signal, cross-modulation arise when its **modulation** is *transferred* to another frequency component [IEC18a]. Amplitude cross-modulation is likely to happen in any non-linear system, however, angle cross-modulation may happen only in a dynamic non-linear system [Raz12, p. 21] — *i.e.*, a system which depends on the past values of its input or its output.

<sup>15</sup> For completeness, intermodulation is also sometimes written as “inter-modulation”.

<sup>16</sup> For completeness, crossmodulation is also sometimes written as “cross-modulation”.



**TIME-DOMAIN ANALYSIS** Amplitude cross-modulation can be easily modeled on a static non-linear system [Raz12, p. 20]. Building on top of our previously introduced non-linear system polynomial in Eq. (2.14), let us consider this input signal:  $x(t) = A_1 \cos(\omega_1 t) + A_2 \cos(\omega_2 t)$ . Using the  $A_1 \ll A_2$  approximation for simplicity, the third-order characteristic of the polynomial would be defined as Eq. (2.19):

$$y(t) = \left( \alpha_1 + \frac{3}{2} \alpha_3 A_2^2 \right) A_1 \cos(\omega_1 t) + \dots \quad (2.19)$$

Let us now suppose that the 2<sup>nd</sup> frequency component of  $x(t)$  is now an interferer which is amplitude-modulated by a **baseband** signal  $m$ . Our input is now defined as Eq. (2.20):

$$x(t) = A_1 \cos(\omega_1 t) + A_2 [1 + m \cos(\omega_m t)] \cos(\omega_2 t) \quad (2.20)$$

Therefore, substituting this new  $x(t)$  into Eq. (2.19) gives us:

$$y(t) = \left[ \alpha_1 + \frac{3}{2} \alpha_3 A_2^2 \left( 1 + \frac{m^2}{2} + \frac{m^2}{2} \cos(2\omega_m t) + 2m \cos(\omega_m t) \right) \right] * A_1 \cos(\omega_1 t) + \dots \quad (2.21)$$

From this equation, we can now observe that the  $\omega_m$  **baseband** signal, originally modulating the  $\omega_2$  frequency component of  $x(t)$ , is now also modulating the  $\omega_1$  frequency component in amplitude.

### 2.2.7 Crosstalk

**Crosstalk (XT)** is a term that is used in a lot of different situations in both **electromagnetic compatibility** and **electromagnetic security** — often, without proper definition. In this section, we focus on the **electromagnetic compatibility** perspective, while the **electromagnetic security** perspective of **crosstalk** will be introduced in Section 3.2.1. The IEC [IEC92] defines **crosstalk** as the following:

The appearance of undesired energy in a channel, owing to the presence of a signal in another channel, is caused by, for example, induction, conduction, or non-linearity.

As such, **crosstalk** is defined as unwanted energy in a channel, being the *consequence* of either a coupling phenomenon (conducted or induced, but not radiated) or a non-linearity product. This definition should remind the reader about the definition of *interference* (in this case, the **crosstalk**) and *disturbance* (in this case, the resulting signal of the coupling or non-linear phenomenon) (introduced in Section 2.2.1). Therefore, one may conclude that the **crosstalk** term is not informative. Nevertheless, some authors define **crosstalk** as a coupling through the

**near-field**, *i.e.*, an *induced* either *inductive* or *capacitive* coupling [Pau06, ch. 19, p. 559] [Devo8a] [Ros82, p. 23]. In this case, **crosstalk** is the phenomenon itself. To conclude about **crosstalk**, even the EMC community uses this term with different meanings — therefore, it is strongly indicated to always define what **crosstalk** means when using it.

## 2.3 ADVANCED ENCRYPTION STANDARD (AES)

**OVERVIEW** **Advanced Encryption Standard (AES)** is symmetric cryptosystem defined by the NIST as the FIPS-197 [SN01]. It is a block cipher, *i.e.*, processing blocks of data to encrypt plaintext into a ciphertext – or the opposite for decryption. **AES** defines several modes, which define how **AES** will handle the different blocks of plaintext larger than its block size. In ECB-128 mode, each block of 128-bit input data is processed separately during 10 iterations, called “rounds”. Each round repeats several operations, *e.g.*, XORing, S-Boxes, and shifts. It can be used with different key sizes (between 128 bits and 256 bits), and the number of rounds depends on the key size (*e.g.*, 10 rounds for the 128-bit key size).

**OPERATION** First, the algorithm will derive different keys for each round using the *key expansion routine* — a combination of S-Boxes (“SubWord”) and cyclic permutation operations (“RotWord”). Second, the next step is to create the so-called *State* by mapping the plaintext  $p$  into a two-dimensional array of 16 bytes in total ( $4 \times 4$  bytes) — for a 128-bit block size. Third, the state is modified according to the “AddRoundKey” operation between the State (containing the plaintext) and the first key — *i.e.*,  $p \oplus k_0$ . Last, each round repeats all following operations (in this order) that will modify the State, except the last round which does not repeat the “MixColumns”:

- *SubBytes*: Apply the S-Boxes to each byte of the internal state, being the only non-linear operation.
- *ShiftRows*: Cyclically shifts the last three rows in the internal state.
- *MixColumns*: Operates on the internal state column-by-column.
- *AddRoundKey*: XORs each column of the internal state with a word from the key expansion.

**SIDE-CHANNEL ATTACK** **AES** has been a target for side-channel attacks for two decades, *e.g.*, with **simple power analysis (SPA)** targeting the key expansion algorithm [Man03], **differential power analysis (DPA)** targeting the first “AddRoundKey” operation [Ors+04], or a timing attacks targeting the T-Table implementation [Ber05]. In this thesis, we leverage the well-known divide-and-conquer approach by attacking

the first round key  $k_0$  byte per byte. The measured leakage is the [electromagnetic radiation](#) of the S-box output, depending on the known plaintext  $p$  and the first round key  $k_0$ , therefore defining our leakage variable  $y$  as in Eq. (2.22):

$$y(p, k) = \text{SBox}(p \oplus k_0) \quad (2.22)$$

In this thesis, we study two types of leakages, known as conventional or Screaming Channels (introduced in Section 3.5.1.1), corresponding to the [near-field](#) or to the [far-field](#), respectively. For the conventional leakage, the [Hamming weight](#) model is a good assumption about the relation between the variable's values and the leakage. However, as studied by Camurati *et al.* [[CFS20](#), p. 372], the leakage is distorted in the Screaming Channels leakage. Therefore, we use a leakage model that is learned from measurement for all the 256 possibilities of the  $p \oplus k$  variable — *i.e.*, a template, introduced in Section 3.4.3.



# 3

## STATE OF THE ART

COMPUTER SECURITY includes offensive and defensive approaches, to defeat or improve a security system. In the offensive context, when an attack exploits a flaw in a computer science-defined protocol, the analysis of the problem is more often straightforward compared to other attacks. However, when an attack exploits phenomena studied by another sensitive field, we observe a “literature barrier” problem. This problem arises when two different fields are describing the same effect or resolving the same problems, but with different terminology. As a result, researchers in both fields are unaware of each other’s research.

Building on our background presented in Chapter 2, we try to tackle this problem by carefully defining the terminology that we will employ in this manuscript. Moreover, when necessary, we will propose alternative names or classifications in the light of our knowledge. From those definitions, we will dig into computer security towards [electromagnetic](#) attacks, mainly through side-channel analysis.

### 3.1 FUNDAMENTAL TERMINOLOGY

Before diving into the state of the art concerning electromagnetic attacks and side channels, we find it interesting to “take a little detour” through basic terminology — so that there is no confusion about what we are talking about.

#### 3.1.1 Introduction of Security Terms

WARE PUBLICATION In 1967, Willis Ware published the first publicly known computer security paper untitled *Security and Privacy in Computer Systems* [War67]. In this publication, Ware discussed information leakage in resource-sharing computer systems and introduced various computer security definitions. Among them, the more important ones are:

PRIVATE Intended for or restricted to the use of a particular person or group.

PRIVACY Isolation, seclusion, or freedom from unauthorized oversight or observation.

At that time, it was common to use different terms for military and non-military information. In this paper, “security” was reserved for military

or governmental information, while “privacy” while used for civilians. Today, a shift in the terminology happened, since security is used for both. Concerning hardware security, Ware speculated on the threats posed by radiation and [crosstalk](#) without further demonstrations. To our knowledge, this is the first public appearance in computer security of those terms — which are still used today in [electromagnetic](#) security research, sometimes without accuracy.

### 3.1.2 *The Multiple Definitions of “Leak”*

The “leak” or “leakage” words are used with multiple different meanings in the literature. Intuitively, it corresponds to a leak of compromising information — a secret. However, there are papers in which this word has a special (and sometimes implicit) definition. Even if we also fall into this pitfall, we found useful to recall the different meanings:

**LEAK VARIABLE** In a side-channel attack context, it corresponds to the intermediate value of a cryptographic algorithm that is modeled.

**LEAKAGE MODEL** In a side-channel attack context, it corresponds to the mathematical model (*e.g.*, the [Hamming weight](#) for a simple one) providing the relation between the *leak variable* and the physical quantity that is measured.

**LEAK SIGNAL** In an [electromagnetic](#) attack context, it corresponds to the sampled [electromagnetic wave](#) which is carrying the compromising information — which can be a [baseband](#) or a [passband](#) signal, as detailed in Section 3.2. In this thesis, when using the term “leak”, it refers to a leak signal — without any other specific precision.

**TRACE** In a time-domain side-channel attack, the “leak” is sometimes a synonym for the real-valued vector being the trace exploited by the attack algorithm.

## 3.2 COMPROMISING EMANATIONS AND EMSEC

**COMPROMISING EMANATIONS** In their most general physical forms, compromising emanations can be defined as energy measured in Joule ( $J$ ) transferred from one system to another, which depend on a piece of information related to a secret. While compromising emanations also encompass energy transferred through mechanical work as acoustic waves, in this thesis, we are focusing on energy transfer described through the theory of electromagnetism. Depending on the coupling mechanism (introduced in Section 2.2.5), the energy may be transferred in the form of electricity, capacitance or inductor, or radiated

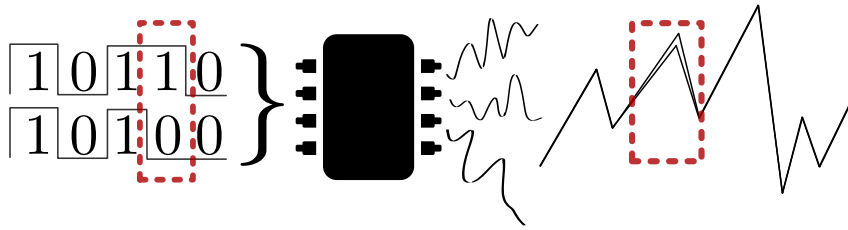


Figure 3.1: A difference in the secret input of an IC will produce a difference in its physical emanation – being compromising.

in the form of a plane [electromagnetic wave](#) (also often mentioned as [radio-frequency wave](#)).

*NACSIM 5000: TEMPEST Fundamentals* [Ros82], the NSA document declassified in 2000, gives the following definition of compromising emanations:

Compromising emanations consist of electrical or acoustical energy unintentionally emitted by any of a great number of sources within equipment/systems which process national security information. This energy may relate to the original message, or information being processed, in such a way that it can lead to recovery of the plaintext.

**TEMPEST** *TEMPEST*, in this general meaning, is the NSA code-name for the program of studying compromising emanations in both offensive and defensive directions. Later in the state of the art, we will see that today, TEMPEST is often associated with passive [EM](#) eavesdropping known as Van Eck Phreaking. The computer security field of studying compromising emanations is also denominated as [Emission Security \(EMSEC\)](#) [Ros82, p. 29], which seems more accurate than TEMPEST.

**EMISSION SECURITY (EMSEC)** Emission Security is the study of compromising emanations in computer security. Before diving into the state of the art of research, we will distinguish several criteria for different types of compromising emanations. From those criteria, we will be able to classify accurately the state of the art, to better understand the relationship between the different attack or defense mechanisms.<sup>1</sup>

### 3.2.1 Sensitivity

The criticality level of information contained in a signal is used to classify two levels of signal sensitivity [Lav+21, p. 2], first defined in the *NACSIM 5000* document [Ros82, p. 30]:

<sup>1</sup> In this section, we are not only listing criteria from the existing literature, but also proposing our own and most accurate classification criteria as well as their definitions.

- *Red signal*: Contains sensitive information or depends on (*i.e.*, is correlated to) sensitive information such that an attacker will be able to retrieve the sensitive information by intercepting a red signal. It is opposed to a black signal.
- *Black signal*: Contains (or depends on) non-sensitive information, or encrypted sensitive information (*i.e.*, a ciphertext). It is opposed to a red signal.

From those definitions, we can apply this terminology to transmission lines that are used to transmit those signals:

- *Red line*: Communication channel intended to transfer a red signal. It is opposed to a black line.
- *Black line*: Communication channel intended to transfer a black signal. It is opposed to a red signal.

From those definitions, we can now define the meaning of the term *crosstalk (XT)* used in the **Emission Security (EMSEC)** context [Lav+21]:

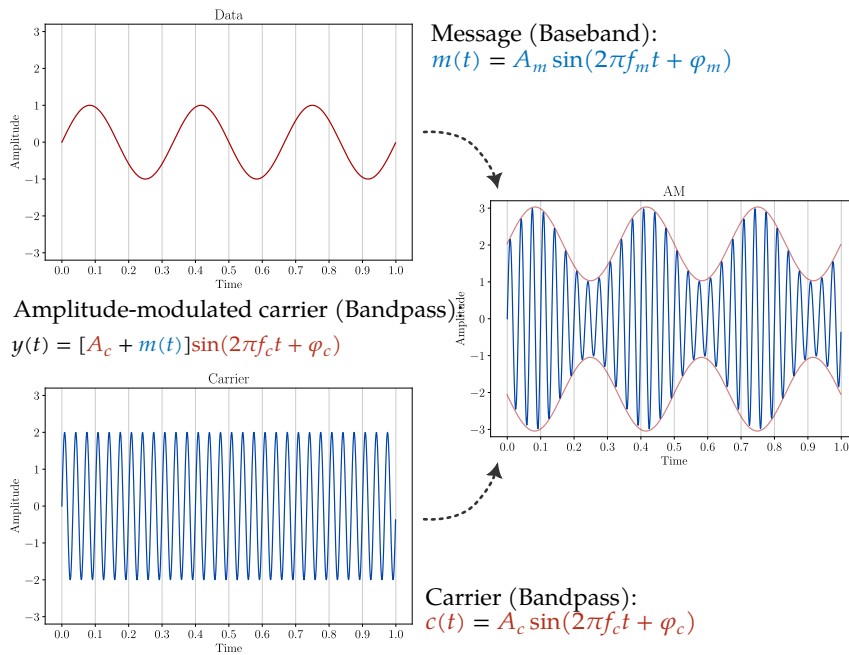
**Definition 3.2.1 (Crosstalk (XT) (EMSEC))**. In the **EMSEC** context, *crosstalk* refers to the consequence of the transfer of a red signal from a red line coupled to a black line.

Historically in communication security, the term *crosstalk* has been mainly used for information leaked from one telephony cable to another. Nowadays, in telecommunication, this type of interference is designated as *co-channel interference (CCI)*. Note that this definition of *crosstalk* (as a consequence) in **EMSEC** is different from the definition of *crosstalk* (as a phenomenon) in **electromagnetic compatibility (EMC)** (introduced in Section 2.2). In 2017, Su *et al.* [Su+17] published an exploitation of *crosstalk* in USB hubs. In this case, the *crosstalk* term usage is twofold: (1) It is valid from an **EMSEC** point of view, since we have a red signal (data of one USB port) which is transferred (and distorted) to a black line (another USB port). (2) It is valid from the **EMC** point of view, since the transfer phenomenon is a capacitive coupling between two lines across the power supply line.

### 3.2.2 Directness

The directness of the emanations depends on the number of “individual signals” in the emanations, *i.e.*, if the sensitive information is modulating another signal or not. It is also a pertinent criterion to depict two categories of compromising emanations since it will affect the frequencies and the distance at which the emanations can be received. Moreover, this criterion is useful to distinguish attacks exploiting similar or different electric phenomena responsible for the **compromising emanation**. Based on both old and recent literature, we will depict *direct*





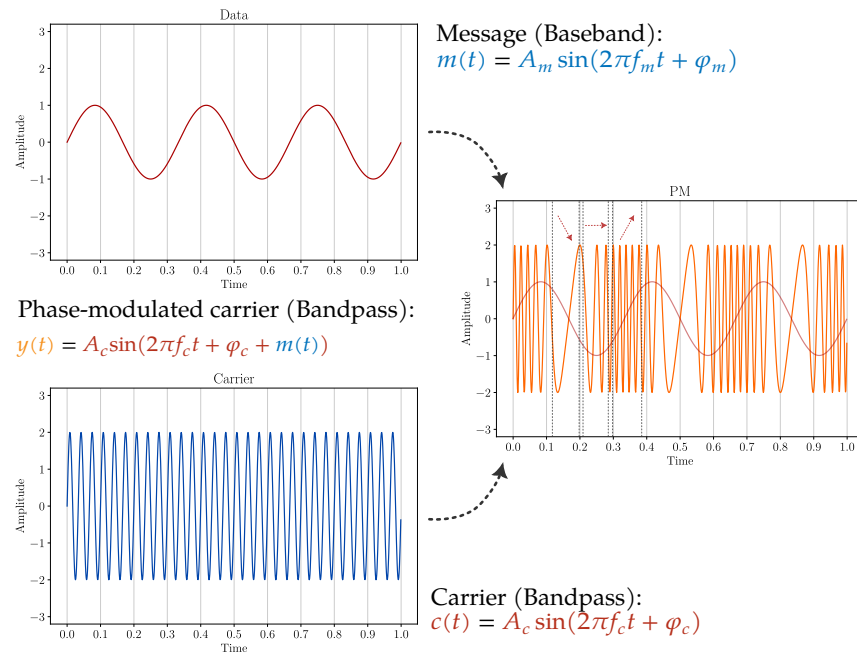
**Figure 3.2:** Illustration of a carrier being modulated in amplitude by a message (red) signal. As a result, the amplitude of the modulated signal is varying according to the value (instantaneous amplitude) of the message signal.

and *indirect* emanations [Ros82; Agr+03; LMM05; VP09; CYC20, p. 8, 4, 282, 3, 1009].

**DIRECT EMANATIONS** In direct emanations, there are no modulation processes. Foundational analysis and exploitation of these emanations are publications from *Quisquater et al.* [QS01] or *Gandolfi et al.* [GMO01], exploiting direct emanations of smart cards. We distinguish between emanations in which the received signal corresponds to the baseband signal or its derivative:

- *Baseband signal:* Emanations corresponding to the red signal. In NACSIM 5000 terminology, it is called *RED Baseband Signals* [Ros82, p. 8].
- *Derived signal:* Emanations corresponding to the energy emitted during the switching activity due to the sharp transitions, *i.e.*, the transition between two logic states of different voltage. The signal characteristics will relate to the switching frequency and the duration of the rising and falling edges of the digital square wave. In NACSIM 5000 terminology, it is called *Impulsive Emanations* [Ros82, p. 9].<sup>2</sup>

<sup>2</sup> For a graphical example, the reader may refer to Figure 1-4 and Figure 1-1 from NACSIM 5000 [Ros82].



**Figure 3.3:** Illustration of a carrier being modulated in phase by a message (red) signal. As a result, the frequency of the modulated signal is varying according to the variation of the value (instantaneous amplitude) of the message signal.

**INDIRECT EMANATIONS** In indirect emanations, there is an unintentional modulation of a carrier by a red **baseband** signal. This modulation may be of various type, as partly illustrated by Fig. 3.2 for amplitude modulation and Fig. 3.3 for phase modulation. Foundational analysis and exploitation of these emanations are publications from *Agrawal et al.* [Agr+02; Agr+03], exploiting both direct and indirect emanations of smart cards. We distinguish two types of carriers being modulated:

- *Unintended carrier:* The carrier is a signal which will act as a carrier but is not intended to be used as a carrier. Such examples are clock signals, commonly exploited in **EM** attacks.<sup>3</sup> In survey from *Lavaud et al.* [Lav+21], this is called *Radio radiation*.
- *Intended carrier:* The carrier is a **RF** carrier that is intended to be used as a carrier and emitted over-the-air, hence, is a **pass-band** signal. Such examples are a carrier output of a **voltage-controlled oscillator (VCO)** sent to an antenna inside an **mixed-signal system-on-chip (MSoC)**, which is exploited by Screaming Channels [Cam+18]. In a survey from *Lavaud et al.* [Lav+21], this

<sup>3</sup> For a graphical example, the reader may refer to Figure 1-3 from NACSIM 5000 [Ros82].

is called *Forced broadcast*.<sup>4</sup> This class of attacks includes passive or active **illumination** attacks and **Screaming Channels**-like attacks.

We also distinguish two intended carrier sources:

- *Internal intended carrier*: The intended carrier is generated internally by the victim itself, *i.e.*, **Screaming Channels** attack.
- *External intended carrier*: The intended carrier is generated externally by an attacker or by a legitimate device in the vicinity, *i.e.*, active or passive **illumination** attack, respectively.

Indirect emanations are also called *Carrier Coupling Emission (CCE)* by Choi *et al.* [CYC20, p. 1009]. In NACSIM 5000 terminology, indirect compromising emanations are called *Modulated Spurious Carriers* [Ros82, p. 8]. The **modulation** type (amplitude or angle) depends on the electronic component which is affected by the coupling effect (introduced in Section 2.2.5) or the non-linearities (introduced in Section 2.2.6). As non-exhaustive examples, a coupling to:

- A **CMOS**' transistor gate line coupled to a baseband signal can act as an *amplitude modulator* [LMM05].
- A **VCO**'s voltage control line coupled to a baseband signal can act as an *angle modulator* [LMM05].

While the latter are the most accepted definitions, “direct” and “indirect” emanation definitions may vary between authors because of improperly defined **EMC** concepts. For the sake of completeness, some authors define emanations from the point of view of the *electronic device designer*. By doing so, *direct* is sometimes called *intentional* and *indirect* called *unintentional*:

- Direct emanations can be qualified as *intentional* – in the sense of unavoidable – since they are inherent or intrinsic to the time-varying currents in electronics.
- Indirect emanations can be qualified as *unintentional* – in the sense of unanticipated – since they result from unintentional interactions between electronic components.

In this thesis, we will never use the latter two definitions from the point of the electronic device designer concerning the intentionality.

<sup>4</sup> However, the author of this manuscript finds that the term “Forced broadcast” from Lavaud *et al.* [Lav+21] may not be the most accurate one, because “Forced” refers to the activeness and not the carrier type. The two subcategories depicted in the Lavaud survey, *i.e.*, “Illumination” and “Mixing”, might also be not the most accurate ones, since the electronic definition of “mixing” does not seem to match all the attacks in its subcategory and that passive illumination attacks, as well as Screaming Channels attacks, are classified as “Mixing” — whereas the two types of attack are distinct except the “Intended carrier” property.

### 3.2.3 Intentionality

The intentionality depicts two categories of emanations depending on the intentionality of the attacker regarding its generation and emission [Lav+21, p. 4]:

- *Unintentional emanations*: Unintentionally emitted compromising emanation (containing a red signal) which are present in a proper working device. This is the definition of the emanations that are exploited by side-channel attacks (presented in Section 3.5). Attacks exploiting unintentional compromising emanations can be passive or active.
- *Intentional emanation*: Intentionally generated compromising emanations (containing an intentional red signal) which are not present without the generation by the attacker. Depending on the generation technique [ZP14] that the attacker will use, it will be constrained to some modulation (e.g., AM [Wan+16], FM [Gur+14; Prv+17] or LoRa [She+21]), frequency bands (e.g., Wi-Fi [Gur23] or GSM [Gur+15]), some hardware (e.g., USB interface [GME16]) — or may be released from a lot of constraints by leveraging the *Noise-SDR* [CF22] technique. This corresponds to the **electromagnetic radiation** that are exploited by *covert channel attacks*, often used to exfiltrate information in an air-gaped context. Such attacks exploiting intentional **compromising emanation** are *active* by definition, since the attacker has to tamper with the victim to generate the emanations (e.g., by running a malicious software on the victim system<sup>5</sup>).

Using the point of view from the **electromagnetic radiation** generation is useful to understand the trigger origin in different attacks.

### 3.2.4 Activeness

The activeness [Lav+21, p. 4] creates two categories of attack scenario depending on whether the attacker<sup>6</sup> is emitting an **electromagnetic radiation** (i.e., **TX**) or if he is only receiving (i.e., **RX**):

- *Passive attack*: The attacker receives the emanations from the victim without emitting any signal (e.g., a passive side-channel or a passive illumination attack (presented in Section 3.3.2)).
- *Active attack*: The attacker emits a signal, which can be emitted from different sources. If the emission originates from inside the

<sup>5</sup> Generating **electromagnetic radiation** from a software is also referred as Soft-TEMPEST, see Section 3.3.1.

<sup>6</sup> When specifying the attacker, it is not only his physical person but also any malicious software or hardware belonging to him.

victim system, it can be an intentional [compromising emanation](#) used for a covert channel (as seen in Section 3.2.3) produced by a malicious code or a malicious hardware implant. If the emission originates from outside the victim system, it can be a carrier signal to:

1. Tamper with the normal behavior of the device, known as [intentional electromagnetic interference \(IEMI\)](#) in the EM context (e.g., [electromagnetic fault attack \(EMFA\)](#) [Est23]).
2. Force a victim system to re-emit a compromising emanation, known as an active illumination attack (presented in Section 3.3.2).

Using the point of view from the attacker for this criterion is useful when defining threat models. It is not to be confused with the *intentional and unintentional* terms used for *direct and indirect* emanations (as seen in Section 3.2.2).

### 3.2.5 *Escape Medium*

While there is often a combination of different [coupling](#) mechanisms in action when assessing [compromising emanation](#), the escape medium is defined as the medium in which the attacker will be able to measure the signal. As such, the escape medium can be:

- *Conducted emanations*: [Compromising emanation](#) that are conducted through electrical conductors available to the attacker. It encompasses the *Conduction (C)* escape medium from the NACSIM 5000 standard [Ros82, p. 17].
- *Radiated emanations*: [Compromising emanation](#) that are radiated through the free space, either in the near-field or the far-field (introduced in Section 2.2.4). It encompasses the *Electric Radiation (ER)* for the electric field ( $E$ ) and the *Magnetic Radiation (MR)* for the magnetic field ( $B$ ) escape media in the near-field from the NACSIM 5000 standard [Ros82, p. 16]. We speculate that it also encompasses the last redacted escape medium from the NACSIM 5000 standard [Ros82, p. 17], which would correspond to the far-field radiation, as exploited in [Screaming Channels](#) attacks or [illumination](#) attacks.

## 3.3 ELECTROMAGNETIC ATTACKS

In this section, we will review the major [electromagnetic](#) attacks in light of the EMC knowledge (introduced in Section 2.2) and the criteria (depicted in Section 3.2).

### 3.3.1 NSA Code Names

Until now, we have carefully defined the different criteria that are used to understand and classify threats models and phenomena in action. We defined them, as the literature sometimes confuses the different terms and phenomena — *e.g.*, giving two different definitions of crosstalk because seen from either the EMC or EMSEC context). However, while we review the public domain terminology, the governmental, military, and intelligence agencies from the USA also used a lot of different code names. These code names refer to standards that address specific attacks and defenses, which can be understood in light of the different criteria that we introduced above.<sup>7</sup>

In 2001, *David Wagner* [Wag01] discussed the possible meanings of the code names used in declassified documents on the *Cryptography Mailing List*<sup>8</sup>, based on the major work of well-known Cambridge scientists *Markus Kuhn*, *Ross Anderson* and one of its student, *Theodore Marketos*. In the following, we will briefly review the most common code names:

**EMSEC** In the NSA terminology, the unclassified EMSEC term (also introduced in Section 3.2) encompasses TEMPEST, HIJACK, NON-STOP standards [AK99; Han99] — for both offensive and defensive techniques.

**TEMPEST** We introduced the general meaning of TEMPEST in recent usage in Section 3.2. However, its historical usage may differ between authors, since its definition has been unclassified only in December 2000 [Ros82]. The author of this manuscript finds it interesting to cite reference definitions:

- From *Anderson and Kuhn* [AK99]:
 

Interception of stray information bearing RF emissions from equipment.
- From the *TEMPEST: A Signal Problem* [Age72] declassified document from 2007:
 

Any time a machine is used to process classified information electrically, the various switches, contacts, relays, and other components in that machine may emit radio frequency or acoustic energy. These emissions, like tiny radio broadcasts, may radiate through free space for considerable distances — a half mile or more

<sup>7</sup> While researchers or journalists find acronyms *a posteriori* (*e.g.*, Telecommunications Electronics Materials Protected from Emanating Spurious Transmissions for TEMPEST), code names are chosen randomly by intelligence agencies such that it does not reveal any information about the attack or defense.

<sup>8</sup> The mails of the mailing list can be browsed at: <https://datwww.mit.edu/bloom-picayune/crypto/4>

in some cases. Or they may be induced on nearby conductors like signal lines, power lines, telephone lines, or water pipes and be conducted along those paths for some distance — and here we may be talking of a mile or more. When these emissions can be intercepted and recorded, it is frequently possible to analyze them and recover the intelligence that was being processed by the source equipment. The phenomenon affects not only cipher machines but any information-processing equipment.

In summary, as *Markettos* [Mar11] emphasized, TEMPEST can be understood as, for the offensive part, the interception of **compromising emanation** and its exploitation, and as the countermeasures against those **compromising emanation** for the defensive part — including both acoustic and electromagnetic emanations. In the modern classification introduced before, in its offensive understanding, TEMPEST would encompass *passive* attacks targeting *unintentional emanations*.

**HIJACK** The term is classified [Air98b, p. 105] [Air98a, p. 22], however, the meaning is speculated by two reference documents:

- From *Markettos* [Mar11]:  
Modulation of secret data onto conducted signals.
- From *Anderson and Kuhn* [AK99]:  
Interception of sensitive information that has somehow contaminated an electrical signal accessible to an attacker (*e.g.*, a power line or ciphertext feed).

Based on speculated meanings, in the modern classification introduced before, in its offensive understanding, HIJACK would encompass *passive* attacks targeting *unintentional emanations* propagating through a *conducted* escape medium. Hence, the attacker would have to receive the signal using direct electrical contact with the conductor (such as the conductive **coupling** mechanism).<sup>9</sup> While TEMPEST is generally understood as intercepting **compromising emanation** through the free space as an escape medium, the TEMPEST standard in fact also consists of testing against propagation across conductors [Ros82, p. 18]. As such, speculated HIJACK meaning would be a subset of TEMPEST, with a particular emphasis on the study of **crosstalk** consequences (as defined in Section 3.2.1), *e.g.*, in power lines.

**NONSTOP** The term is classified [Air98b, p. 105] [Air98a, p. 22], however, the meaning is speculated by two reference documents:

<sup>9</sup> Note that in the end, the conductor may radiate or transfer the red signal to another radiating conductor.



- From *Markettos* [Mar11]:  
Modulation of secret data onto radiated signals.
- From *Anderson and Kuhn* [AK99]:  
Interception of sensitive information that has accidentally modulated secondary emissions of an RF carrier such as a mobile phone or radar signal.

Based on speculated meanings, in the modern classification introduced before, in its offensive understanding, NONSTOP would correspond to an **illumination** attack, *i.e.*, a *passive* or *active* attack exploiting *radiated emanations* in the far-field from the victim, in which the **compromising emanation** is an *intended RF carrier* modulated by a red signal.<sup>10</sup> **Illumination** attacks will be introduced in more detail in Section 3.3.2.

**TEAPOT** The term has been unclassified since 1999, found in the *NSA/CSS Regulation: Reg. No. 90-6* [Der91, p. 8], but its original date is partially redacted (“199x”):

A short name referring to the investigation, study, and control of intentional compromising emanations (*i.e.*, those that are hostilely induced or provoked) from telecommunications and automated information systems equipment.

*Markettos* [Mar11] also gives the following definition:

Intentional malicious emissions.

In the modern classification introduced before, in its offensive understanding, TEAPOT would encompass *active*<sup>11</sup> attacks targeting *intentional emanations* — *e.g.*, covert channel attacks.

### 3.3.2 *Illumination (Re-Emission)*

**FIRST PUBLICATION** An *illumination* attack, also known as *re-emission* and *re-radiation* attack, has been first introduced in the public domain by *Burnside et al.* [BEA08] in 2008. In its original publication, the attack makes a special usage of **intentional electromagnetic interference (IEMI)**, more precisely, it exploits an actively emitted carrier from the attacker that is re-radiated (or re-emitted) from the victim while being modulated by a data-dependent signal. The authors demonstrate a frequency domain analysis of the side channel, suggesting that **simple power analysis (SPA)** is usable in the time domain without demonstration. It is defined by its authors as:

<sup>10</sup> While never confirmed officially, by carefully reading the US Air Force’s *AFSSI 7010 and AFSSM 7011* [Air98a; Air98b], the speculated meaning seems highly probable.

<sup>11</sup> For TEAPOT, active means that the emissions are due to an attacker controlled malicious software or hardware.



[...] re-emission side channel, which exploits the modulated scattering due to the **illumination** of an integrated circuit by an external RF carrier. [...] It is demonstrated that data-dependent switching behavior is cross-modulated on the introduced carrier, providing an additional side-channel outwith the switching frequency of the device.

However, while the previous literature [Lav+21] used the term *illumination* to designate *active* attacks implicitly, we use the term *active illumination* for an external carrier that is introduced by the attacker and *passive illumination* for an external carrier that is introduced by a legitimate device in the vicinity. Therefore, the active or passive properties are concerns for the attacker threat model, while from the victim's point of view, the phenomenon leading to the leakage is the same. In conclusion, we can define an **illumination** attack as follows:

**Definition 3.3.1** (Illumination). **Electromagnetic (EM)** attack which exploits *unintentional, radiated* and *indirect compromising emanation* with the modulation of an *external intended RF carrier* because of a data-dependant switching activity.

**MILITARY DENOMINATION AND USAGE** As seen in Section 3.3.1, it is highly probable that the *NONSTOP* code name from the NSA refers to an **illumination** attack. Another well-known usage of an *illumination* attack in the military world is *The Great Seal Bug*, also known as *The Thing*, used by the Soviet Union during World War II to eavesdrop audio in an ambassador U.S. room. This device is a passive microphone with a design to create a capacitive **coupling** between a monopole antenna and a resonant cavity (dependent on ambient sound). By illuminating the device and receiving the modulated re-emission, the Soviet Union was able to spy over a distance of 20 meters [RS15].

**ROOT CAUSE** The root causes for the modulation of the reflection of the incident carrier, *i.e.*, the wave re-emitted by the victim, is explained in old and recent papers.<sup>12</sup> As studied by *Burnside et al.* [BEA08], the carrier reflected from the target device is modulated because of impedance variations of the input and output of the CMOS logic (*e.g.*, inverters) — which depends on its logic state (high or low). The scattered field, *i.e.*, the reflected carrier, obeys the same laws as in “standard Radar problem”, where the modulation depends on the terminating impedance of a metallic structure, acting as a modulating retro-reflector [TE98]. *Kaji et al.* in *Echo TEMPEST* [Kaj+23]<sup>13</sup> describes the incident carrier

<sup>12</sup> For further reading about how an incident carrier will penetrate, propagate or reflect, the reader may refer to the Diffusion Penetration [GSH20, p. 76] and the Aperture Penetration [GSH20, p. 80] transfer functions.

<sup>13</sup> The author of this manuscript note that the *Echo TEMPEST* [Kaj+23] publication seems to surprisingly ignore the work done by *Burnside et al.* [BEA08] about illumination, 15 years before.

as “received” by the target device through its “unintentional antennas” (e.g., wires and PCB tracks [Pau06, p. 528]). However, the incident wave is partly absorbed by the target device depending on the impedance of the unintentional antennas, which is varying in time, producing an amplitude-modulated reflected wave. In *Echo TEMPEST*, this modulation is especially produced by the switching activity of the output buffer used by the I/O port of an [integrated circuit](#), leading to the compromise of processed data transferred on this port.

**ACADEMIC PUBLICATIONS** In recent years, illumination attacks have been demonstrated for various purposes in academic publications. In 2015, Wei et al. [Wei+15] exploited an illumination attack to perform a wireless vibrometry, in which they receive the reflected carrier and infer from its modulation (called an Acoustic-Radio Transformation (ART)) the acoustic waves of a loudspeaker to perform an audio eavesdropping. In 2023, Kaji et al. [Kaj+23] exploited an illumination attack very similar to Burnside et al. [BEA08] in order to infer the digital values of input/output peripherals (e.g., demonstrated on keyboards). In 2024, Samy Kamkar presented at DEFCON [Kam24] an attack in which he used a laser to illuminate a computer, received its reflected wave, and reconstructed from this the acoustic waves generated by the keyboard typing, inferring the keystrokes. Compared to Wei et al. [Wei+15], Kamkar used a laser instead of a [radio-frequency \(RF\)](#) carrier. More generally, this work is an original implementation on how to use a laser as a microphone, a so-called “laser microphone” well-known since at least a decade [BB13].

### 3.3.3 *Soft TEMPEST*

**ORIGINAL PUBLICATIONS** *Soft TEMPEST* is the concatenation of “Soft” (as in “Software”) and “TEMPEST” (which study unintentional compromising emanations) to designate the usage of software to influence or generate [electromagnetic radiation \(EMR\)](#). It was first introduced in this offensive usage by Kuhn and Anderson [KA98] in 1998, in which malicious software is used to encode secret information inside software-controlled [electromagnetic radiation \(EMR\)](#). Then, a defensive publication by Kuhn and Anderson [AK99] in 1999 destined to NATO depicts the usage of the software as a countermeasure against [compromising emanation](#), by leveraging, e.g., like filtering or randomization.

**SIGNAL GENERATION** A research area emerged from *Soft TEMPEST*, studying how to *intentionally* produce [electromagnetic radiation](#) by using signal generation from software. Several publications from Callan, Zajic, and Prvulovic [CZP14; ZP14; CZP15; Wan+16; Prv+17] developed basic software techniques using alternating patterns to produce amplitude-modulated and frequency-modulated unintentional [electro-](#)

magnetic radiation.<sup>14</sup> Zajic and Prvulovic [ZP14] proposed an analysis on how electromagnetic signal can leak information — studying the source and modulation process. This work demonstrates how to exploit the leakage by leveraging micro-benchmarks, alternating two activities to create an amplitude modulation (AM) signal (more precisely, an On-Off Keying (OOK) modulation) at a specific frequency. A more advanced work is *LoRa Meets EMR* [She+21], where the authors succeed in modulating DRAM buses unintentional emanations with a LoRa modulation scheme, leveraging the Spread Spectrum Clock (SSC) feature of DRAM clocks. The latest work in this area are *Noise-SDR* from Camurati et al. [CF22] and *SideComm* [Fen+23] from Feng et al., published in 2022 and 2023 respectively. In *Noise-SDR*, digital modulation techniques are used to produce arbitrary modulated electromagnetic radiation (including LoRa) from an atomic binary operation (i.e., “on-off”), acting on hardware that is known to be a leakage source acting as an unintentional antenna (e.g., DRAM buses). In *SideComm*, CPU operations are characterized and then used as the leakage source. A well-known proof of concept of Soft TEMPEST in the public domain is *Tempest for Eliza* from Erik Thiele [Thio1] published in 2001, which uses intentional electromagnetic radiation from CRT monitors to send a signal amplitude-modulated by the content of a music file.

**COVERT CHANNEL** Covert channels are an application of signal generation from Soft TEMPEST in order to exfiltrate (often, secret) data. The software will inevitably leverage a hardware component in an unintended way to perform the intentional electromagnetic radiation. For example, *LoRa Meets EMR* [She+21] and publications from Guri et al. [Gur23; Gur+15] leveraged DRAM buses to act as an unintentional antenna and exfiltrate data. In a slightly different way, Guri et al. also exploited USB buses [GME16] and Cui [Cui15] exploited lines connected to GPIO to generate compromising emanation for data exfiltration. Another example is SpctrEM [Meu+23], which use the characterization of the unintentional electromagnetic radiation of different CPU operations to perform a covert channel using intentional electromagnetic radiation during the Spectre attack transient window.

### 3.3.4 Van Eck Phreaking

**PUBLICATIONS** Originating from the work of Van Eck [Van85], the term *Van Eck Phreaking* is associated with eavesdropping through unintentional compromising emanation. Sometimes called TEMPEST attack

<sup>14</sup> Alenka Zajic and Milos Prvulovic published *Understanding Analog Side Channels Using Cryptography Algorithms* in 2023 [ZP23], giving a detailed overview of modeling, generating, and analyzing software generated electromagnetic radiation and their applications to perform cryptographic side channels and hardware trojan detection.

by researchers or journalists<sup>15</sup>, this work demonstrates the exploitation of [compromising emanation](#) to eavesdrop on a cathode-ray tube (CRT) monitor. *Kuhn* extensively studied *Van Eck* work and its applications during his *Ph.D.* [[Kuh03](#)] in 2003, including an extension on liquid-crystal display (LCD).

**IMPLEMENTATIONS** Several public implementations exist for monitor eavesdropping:

- *TempestSDR*<sup>16</sup>: A well-known demonstration of monitor eavesdropping software was first released by *Marinov* during his master degree [[Mar14](#)] in 2014. This is a Java graphical application implemented over a C library that allows to perform monitor eavesdropping using a [software-defined radio \(SDR\)](#).
- *gr-tempest*<sup>17</sup>: The second public implementation has been done by *Larroca* [[Lar+22](#)] published in an academic paper in 2022. Inspired from *TempestSDR*, it is a re-implementation with improvements using a collection of GNU Radio blocks implemented in C and bonded to Python. One of the major improvements is to leverage the frequency-modulation of the unintentional [electromagnetic radiation \(EMR\)](#) instead of only the amplitude of the signals.
- *Deep-Tempest*<sup>18</sup>: The third public implementation has been done by *Fernández, Larroca et al.* [[Fer+24](#)]. It is an improvement over *gr-tempest* using a deep-learning trained neural network to significantly improve the image recovery.

### 3.3.5 Jamming

Jamming corresponds to the deliberate blocking of legitimate radio signals by an attacker using [intentional electromagnetic interference \(IEMI\)](#).<sup>19</sup> Most often, jamming is aimed at blocking the radio signal for a receiver — but it can also be used against an emitter. It is then an *active* attack.

**STRATEGIES** We distinguish several jamming strategies:

- 15 If we refer to the TEMPEST presentation in Section 3.2, we understand that a “TEMPEST attack” is so general that it does not well inform about the nature of the attack.
- 16 *TempestSDR* code is hosted on GitHub at:  
<https://github.com/martinmarinov/TempestSDR>
- 17 *gr-tempest* code is hosted on GitHub at:  
<https://github.com/git-artes/gr-tempest>
- 18 *Deep-Tempest* code is hosted on GitHub at:  
<https://github.com/emidan19/deep-tempest>
- 19 If interested in jamming, the reader is encouraged to read the comprehensive jamming review been done by *Pirayesh, Zeng* [[PZ22](#)] in 2022, addressing both offensive and defensive techniques.

- *Proactive*: In the proactive mode, the jammer is performed following a predefined pattern, either by emitting constantly on a single channel, by sweeping over the different channels, or by choosing random channels.
- *Reactive*: In the reactive mode, the jammer is reacting to an input signal or input trigger to start the jamming. One application of *reactive* jamming is *selective* jamming, where the attacker is jamming only a subset of possible messages to or from a legitimate device. This mode can allow jamming only messages closing a security system (*e.g.*, a garage door) but not messages that open it, leaving and forcing the security system to stay open. The challenges are to detect the message types (*e.g.*, by decoding, pattern matching, or using signal correlation) and to decide whether to jam or not to jam fast enough. If the jammer cannot be fast enough, another strategy is always to jam, then re-transmit the analyzed message if a desired message (*e.g.*, an open message) is captured.

**HARDWARE** With the rise of [software-defined radio \(SDR\)](#), it is now very accessible to perform advanced jamming techniques implementation [SN20]. *JamRF* from Ali, Baddeley *et al.* [Ali+22] is an example of a research project using GNU Radio and [software-defined radios](#) to implement different jamming strategies.<sup>20</sup>

**WAVEFORM** Marin *et al.* [MBR24] compared different signals for jamming performance, including random Gaussian noise or specific waveform. This work concludes that targeting the pilot sub-carriers of the attacked protocol using simple tones (*i.e.*, sinusoidal with fixed frequencies) is the most effective.

### 3.4 SIDE-CHANNEL ATTACKS

A side-channel attack is an attack against a security system using information originating from the interaction between the security system and its environment. Hence, it allows for an attacker to recover the secret key used during a [cryptographic operation \(CO\)](#), *e.g.*, the encryption of plaintext by a symmetric cryptosystem or signing data by a digital signature scheme. To recover the cryptographic material, a side-channel attack exploits a physical measurement that can be performed across several different media. Side-channel attacks originate from cryptanalysis, introduced by Kocher *et al.* in 1996, by exploiting first the time taken by the attacked [cryptographic operation](#) [Koc96] and then its power consumption [KJJ99] in 1999. Despite being known and studied in the

<sup>20</sup> JamRF project is hosted on GitHub at:  
<https://github.com/tiiuae/jamrf>

academic domain for at least 25 years, side channels are important threats that are still leading to security issues. Indeed, no matter the size of the key or the robustness of the cryptographic algorithm (*e.g.*, by respecting the *confusion* and *diffusion* principles of block ciphers<sup>21</sup>), a cryptosystem could be potentially broken by a side-channel attack without specific countermeasures.<sup>22</sup>

**ACTIVENESS** A side-channel attack is achieved by gaining information about the secret information being computed on the attacked hardware. By definition, in a side-channel attack, the attacker is not interfering or tampering with the hardware, which should be in a properly functional state. As such, a side-channel attack is by definition a *passive* attack. On the counterpart, an *active* attack is achieved by modifying the computations of the hardware. Such an example is a fault attack, which consist of injecting a fault, *i.e.*, causing a computational error, during the computation that is attacked. Because of the injected fault, a cryptosystem like DES could leak some information, and then be analyzed through side-channel-like algorithms (*e.g.*, Differential Fault Analysis (DFA) [BS97]). Available methods [Gou05] for inducing faults are, among others, power supply, clock signal, temperature, [radio-frequency intentional electromagnetic interference](#), visible light, or Eddy currents — and can be mapped to the side-channel media, introduced in Section 3.4.2). Fault attacks (active) and side-channel attacks (passive) share some properties. For example, Spruyt *et al.* [SMC20] shows how the results of fault attacks can be transformed into a trace used in a passive side-channel analysis. Another example is from Amiel *et al.* [Ami+07], which shows how an attacker can use a fault attack to disable a side-channel countermeasure, enabling the passive analysis.

**INVASIVENESS** A side-channel attack can be achieved by either modifying or not the attacked hardware:

- *Non-invasive*: Attack which does not modify the hardware that is attacked. This class of attack is much more probable to happen in the wild since it can be performed at a distance (depending on the measured medium).

<sup>21</sup> Identified by Claude Shannon, *confusion* implies that each digit of the ciphertext should depend on multiple parts of the key, while *diffusion* implies that one bit of plaintext modification should change half of the bits of the ciphertext.

<sup>22</sup> For further reading about side-channel attacks, the reader may be interested by the following reference books:

- “Power Analysis Attacks: Revealing the Secrets of Smart Cards” from Mangard, Oswald and Popp [MOP07],
- “Security Engineering” from Ross Anderson [Ros20, ch.19],
- “The Hardware Hacking Handbook” from Colin O’Flynn [OW21, ch.8-13].



- *Invasive*: Attack achieved by modifying the hardware that is attacked. The modification is generally not trivial and could be potentially destructive. Consequently, this category is less likely to be usable in the wild. An example is *probing attack*, which corresponds to the use of specialized probes, inserted directly inside the attacked hardware to be able to gain information about the computations.

### 3.4.1 Dependency

A side-channel attack is gaining information about a secret value because of a correlation between the value and a physical measurement. We distinguish two types of dependency between the secret value and the measurements [MOP07, p. 6]:

- *Instruction flow dependency*: The measurements depend on the instruction flow of the targeted algorithm, which itself depends on the secret value. For example, using [simple power analysis \(SPA\)](#) to reveal the instruction sequence, allowing to identify the bits of the exponent during the RSA exponentiation or to identify the bits of the key during the DES key scheduling [KJJ99].
- *Data dependency*: The measurements directly depend on the value of the data being processed. For example, using [correlation power analysis \(CPA\)](#) to reveal the secret information based on a leakage model such as [Hamming weight](#) or [Hamming distance](#), mapping the different secret values to possible measurements.

### 3.4.2 Media

Since the discovery of side channels, various ways of measuring the side-channel information were discovered [ZF05]. In the following, we will give a high-level overview of different media, or channels, that can be used to gain side channel information:

**ELECTRICAL** [KJJ99] There is no electrical component with a perfectly stable electrical power consumption. Moreover, power consumption is often dependent on the executed instructions and their operand values, *i.e.*, the data. Hence, measuring the power consumption through the voltage drop across an electrical resistor was one of the first media used to gain side-channel information.

**ACOUSTIC** [GST17] Analyzing the acoustic waves, *i.e.*, the “sound” emitted by the attacked hardware could allow an attacker to break a cryptosystem. This corresponds to the illustration of the lock-picker attacking the vault given in Section 1.2.2, with the difference that the acoustic waves emitted by electrical components are not

exploitable without recording and signal processing techniques — since the frequency does not always correspond to a human-audible sound, *e.g.*, ultrasonic sound.

**TIME** [Koc96] Timing information refers to the time taken for a given computation. In fact, depending on the value of some variables, the critical computation could take more or less time. If we can measure the time of the computation, then we can deduce the values of the variables.

**ELECTROMAGNETIC** [QSo1] In the same way as the electrical consumption, the **electromagnetic radiation** of a hardware component is often correlated with the instruction and its operands. Receiving, measuring, and processing them is a powerful vector of side-channel attacks. This is the category of side channels that interests us in the rest of this thesis, detailed in Section 3.5.

**LIGHT** Light corresponds to a specific type of **electromagnetic radiation (EMR)**, at significantly higher frequencies (around  $1 \times 10^{15}$  Hz) than the **radio-frequency electromagnetic wave** (between 3 Hz and  $300 \times 10^9$  Hz). Visible light is commonly exploited by recording images through cameras [Nas+23] or optical equipment [Bac+09]. For example, possible exploitation is when the brightness of the LED of a device is correlated to the power consumption of the attacked hardware [Nas+23]. It is then possible to use the measurement of the LED brightness as a proxy for a power trace. Invisible light can also be exploited. For example, considering the wave-particle duality, Ferrigno *et al.* [FHo8] successfully gain side-channel information by detecting and measuring the photon emanations from the switching activity of the transistors.

### 3.4.3 Overview

**ALGORITHMS** From a high-level perspective, a side-channel attack is conducted by collecting a high number of measurements – called “traces” – with a fixed key and variables plaintexts. From these traces, one can then use simple analysis heuristics or statistical processing to leverage the traces in order to retrieve the secret information. For example:

- Using **simple power analysis (SPA)** [KJJ99], one may infer the secret bits by a visual inspection, depending on whether the samples value of a trace is higher or below a predefined threshold.
- Using **differential power analysis (DPA)** [Koc+11], one may infer the secret bits using a statistical technique. The attacker will separate his measurements into subsets — using a “selection function” based on an intermediate value, average the subsets, and compute a difference trace of these averaged subsets. When a difference



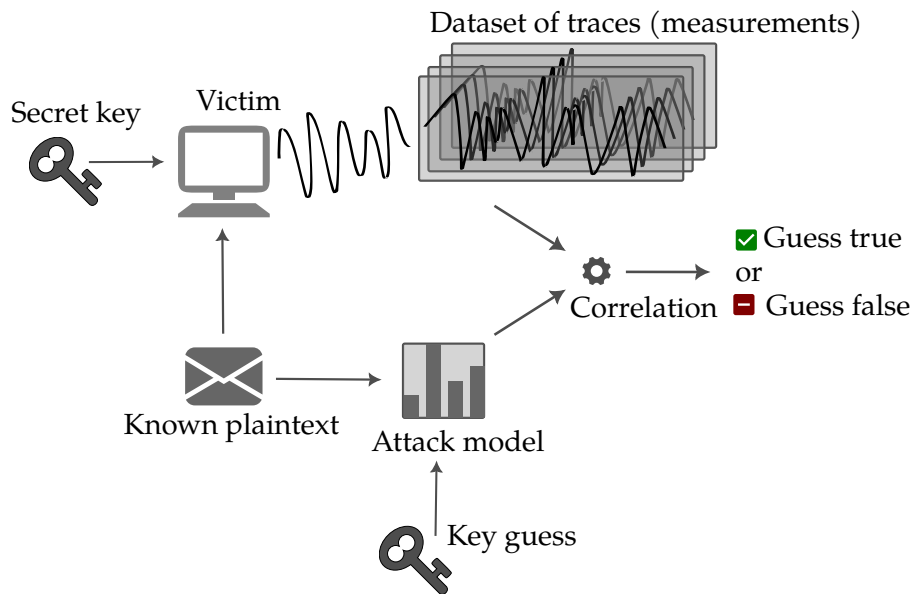


Figure 3.4: Example of a non-profiled correlation based side-channel attack.

trace is significantly different from zero, it allows inferring the secret value depending on the selection function used for this difference trace.

- Using *correlation power analysis (CPA)* [BCO04], one may infer the secret bits by finding the higher correlations between a leakage model and the measurements. The leakage models that have proven to be efficient without being complex are the *Hamming weight* and the *Hamming distance*.

**MODELS** When choosing a *correlation-based* attack – such as *CPA* and its derivatives, the *leakage model* predicts the leakage measurement of an intermediate value used in the *cryptographic operation* based on theoretical data input to the cryptosystem. We distinguish two types of model creation:

- *Non-Profiled Attacks*: The model is chosen on a mathematical relationship between the secret value and the measurements. An example is the *Hamming weight*, where the leakage would be proportional to the number of high logic states in the processed secret. Another example is the *Hamming distance*, where the leakage would be proportional to the number of different logic states when switching from one intermediate secret value to another. The general workflow of a non-profiled correlation based attack is illustrated in Fig. 3.4.
- *Profiled Attacks*: Prior to the attacker, the model is estimated by collecting a high number of training measurements, in order to

build a profile or a *template* [CRR02]. From an information theory consideration, this attack is the most optimal one.

The **Pearson correlation coefficient (PCC)** ( $\rho$  or  $r$ ) is often used as a distinguisher, the function used to compute the correlation between the leakage model and the measurements. The PCC can also be used to find **points of interest (POIs)** [Mey12], the time samples where the leakage is located. Searching for the candidate key with the best correlation, the side-channel algorithm may lead to a full key recovery if the number of traces is sufficient and the noise is not predominant.

**MEASUREMENT REQUIREMENTS** The side channel can be modeled by Equation 3.1 [Oul+20], where the side-channel attack targets the cryptographic intermediate value:

$$\begin{cases} Z(t, k) \rightarrow Y(Z) \\ X = Y(Z) + N \end{cases} \quad (3.1)$$

where:

- $Z$  Cryptographic intermediate value.
- $t$  Known plaintext.
- $k$  Unknown key.
- $Y$  Leakage corresponding to  $Z$ .
- $X$  Leakage measured by an attacker.
- $N$  Gaussian noise.

When the target device will process the cryptographic value  $Z$ , this will produce a related leakage  $Y$  in the physical environment. We observe that the measurement  $X$  is composed of  $Y$  plus some independent and random noise  $N$ . The most important observation for our work is that the relation between  $Z$  and  $Y$  is deterministic. From this equation and previous introduction of side-channel attacks, we can highlight two measurement requirements for conducting an attack:

1. The measured trace is a 1D real-valued vector because side channels are usually performed in the time-domain — where the trace represents the measured physical quantity over time.
2. For a given value of a given cryptographic inputs ( $t$  and  $k$ ), the theoretical leakage  $Y$  will be constant — but the measured leakage  $X$  will still be affected by a degree of randomness because of the noise.

Those two requirements will become challenges when analyzing the phase in the side-channel context, described in Section 8.2.1 from Part III.

### 3.4.4 Multi-Channel Attacks

Agrawal *et al.* [ARR03] were the first to introduce the concept of *multi-channel attack*. At first, it designates the use of multiple channels in a single attack, *e.g.*, exploiting the power and [electromagnetic radiation](#) source simultaneously. Since then, it has been generalized to multiple sources of information which are combined to increase the performance compared to a single side-channel attack. Multi-channel attacks are to side channels what diversity is to radio communications. Yang *et al.* [Yan+17] systematized multi-channel attacks, also called *fusion* or *combination* depending on the context, or using the [multi-channel fusion attack \(MCFA\)](#) acronym. [Yan+17] classified MCFA algorithms into 3 categories that depend on the level at which a chosen combination function is applied to merge the channels:

- *Data-level*: The combination function is used to merge the samples of the traces. An example of a combination function would be the average of two samples for each point in time.
- *Feature-level*: The combination function is used to merge features extracted from the samples.
- *Decision-level*: The combination function is used to merge the final result of independent attacks on each channel. An example of a combination function would be the average of two correlation coefficients for each possible guess on the independent channels.

Genevey-Metat *et al.* [GGH19] used multi-dimensional machine learning algorithms to perform a data-level combination. Meynard [Mey12] used the decision-level combination by using the product of the [PCC](#) of two [points of interest \(POIs\)](#) at different times — allowing a form of time diversity. Mather *et al.* [MOW14] used another method of decision-level combination by combining several intermediate value targets in the cryptographic algorithm — allowing of form of leakage variable diversity.

### 3.4.5 Evaluation Metrics

A performance evaluation evaluates the ability of the chosen side-channel algorithm to correctly retrieve the secret (*e.g.*, a secret key). In side channel attacks, the attacker is often using a divide-and-conquer strategy, where the key is divided into subkeys. For example, in AES, each subkey corresponds to a byte of the key and are attacked independently. Simple methods exist to evaluate the performance of the attack, *e.g.*, computing the [Hamming distance](#) between the correct and the guessed secret. However, this is far from an ideal evaluation metric, since the side-channel algorithm does not output a single value – but

a list of possible guesses, from the most probable to the least probable. Therefore, in this thesis, we will use two metrics that address this problem.

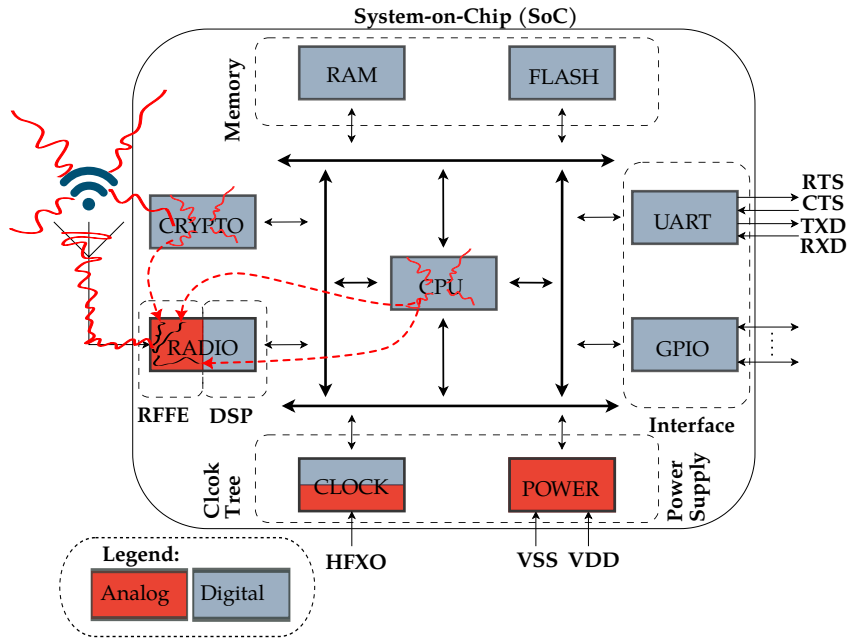
**PARTIAL GUESSING ENTROPY (PGE)** [PDY16] The PGE is defined by the rank (*i.e.*, the index) of the correct subkey among a list of all possible subkeys classified from the most probable to the least probable according to the side-channel output. For a single subkey, it hence estimates how many guesses are needed to find the correct subkey. The “*Partial*” refers to the fact that we are considering only a single subkey instead of a complete key. Hence, a PGE equal to zero indicates that the subkey has been correctly found by the side-channel output.

**KEY RANK (KR)** [VGS13; Glo+15] The key rank defines the rank (*i.e.*, the index) of the correct key among a list of all possible keys classified from the most probable to the least probable. Estimating how many guesses are needed to find the correct key is representative of the complexity of the key recovering *via* a brute-force attack after a side-channel attack. The key rank enumeration is a key brute force leveraging knowledge of the side-channel output. However, such a list is only theoretical – the side-channel algorithm attacks each subkey separately. Hence, algorithms have been developed to estimate the key rank (key rank estimation) instead of brute-forcing the key (key enumeration) to find the key rank if the key is found or its lower bound if the key is not found. We are using the *Histogram-Based Enumeration Library (HEL)* from Poussier *et al.* [PSG16], using both key rank estimation and key enumeration.<sup>23</sup> Performing a key enumeration is an offline procedure that happens after the side-channel attack. On our machine based on an Intel Core i7-11700 with 8 cores and 16 threads at 2.50 GHz, testing  $2^{38}$  keys required approximately 10 hours.

### 3.5 ELECTROMAGNETIC SIDE-CHANNEL ATTACKS

**Electromagnetic radiation (EMR)** has been identified to be a threat against cryptosystems by *Quisquater et al.* [QS01] with a similar impact as power consumption. Numerous algorithms have been developed to exploit this EMR measurement, the first ones being **simple electromagnetic attack (SEMA)** and **differential electromagnetic attack (DEMA)** [Agr+02] which are analogous to **simple power analysis (SPA)** and **differential power analysis (DPA)** [KJJ99] commonly used in power side-channels. EM side channels have been systematically studied by *Lavaud et al.* [Lav+21], describing multiple attack scenarios and systematizing the current literature.

<sup>23</sup> The reader may be interested into a newer key rank estimation methods, such as a Monte Carlo-based instead of Histogram-based proposed in *MCRank* [CDS22].



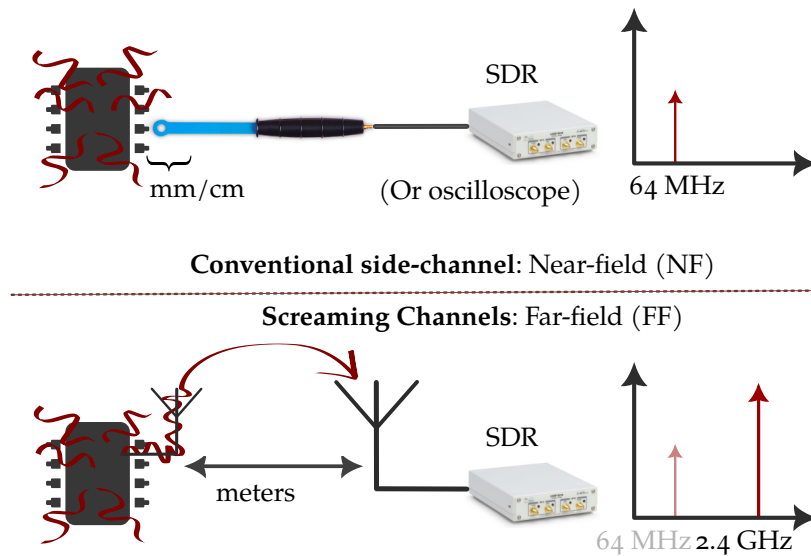
**Figure 3.5:** Coupling path (in red) inside a **system-on-chip** vulnerable to Screaming Channels as studied by *Camurati et al.* [Cam+18]. The conventional **compromising emanation** are still emitted by the computational units (e.g., the CPU and the CRYPTO blocks), but also additionally by the radio transceiver during a radio transmission.

### 3.5.1 Distant Attacks

#### 3.5.1.1 Screaming Channels

Published in 2018 by *Camurati et al.* [Cam+18], the discovery of the Screaming Channel leakage describes how the leak signal propagates through the targeted mixed-signal chip and demonstrates an attack on a custom firmware, *i.e.*, constantly running **cryptographic operations** while transmitting modulated random packets over the air. It concentrates on how far the leak can be exploited in an ideal scenario, increasing the range of traditional side-channel attacks from dozens of centimeters to more than 10 meters. The same authors proposed a more advanced analysis of the leakage in 2020 [CFS20]. This paper evaluates the non-linearity of the best leakage model and proposes to use new statistical tools to increase the attack performance. In Screaming Channels publications [Cam+18; CFS20; Cam20], *conventional leakage*, which is radiative **EMR** recorded using a **near-field (NF)** probe, is clearly distinguished from *Screaming Channel leakage*, which is recorded using an antenna in the **far-field (FF)** after the leakage has been transmitted by a radio transmitter (e.g., due to mixing between the analog and the digital part of the chip) [Cam+18].

*Screaming Channels exploits EM leakage that propagates to (and is broadcast at distance by) the radio transceiver from the digital logic.*



**Figure 3.6:** Comparison of the consequences of a Conventional Side-Channel attack *vs.* a Screaming Channel attack. As a result of the Screaming Channel leakage, the **compromising emanation** (in red) is upconverted to the frequency of the *internal intended RF carrier* and broadcasted at a higher distance.

*Distinguish an Illumination attack from a Screaming Channel attack depends if the RF carrier that is modulated is internal or external to the victim device.*

While studied as a cryptographic side-channel application by the first authors [Cam+18; CFS20], the *Screaming Channels* term corresponds to the modulation of a carrier internally present into the victim device. This may be the result of various electronic phenomena, including substrate coupling (introduced in Section 2.2.5.2 and illustrated in Fig. 3.5) present in some **mixed-signal system-on-chip** as depicted by the original papers [Cam+18; CFS20]. However, Screaming Channels is not exclusive to substrate coupling or **mixed-signal system-on-chip**, we therefore propose the following definition:

**Definition 3.5.1** (Screaming Channels). **Electromagnetic (EM)** attack which exploits *unintentional, radiated and indirect compromising emanation* with the modulation of an *internal intended RF carrier* because of a data-dependant switching activity.

Compared with **illumination** attacks (presented in Section 3.3.2), the specificity of Screaming Channels is that the intended **RF carrier** is internal instead of external.

### 3.5.1.2 Further Extensions of Screaming Channels

Several researchers extended this preliminary work in different directions. Wang *et al.* [WWD20] tried to increase attack performance by using a deep-learning-based approach applied to the original custom firmware, allowing to decrease the required number of traces. Guillaume *et al.* [Gui+22] aimed at reducing the required prior knowledge about the leakage to perform a Screaming Channel attack. As such,

they elaborated a new method to take up the triggering challenge by introducing Virtual Triggering, which aims at finding leakage related to **cryptographic operations (COs)** contained in a trace. From a single trace containing many **cryptographic operations (COs)** over time, the method will isolate each of them in separate segments. It works by finding an approximation of the **CO** duration using autocorrelation and trying different possibilities until finding the precise duration. However, the interest of the method seems limited in a real-world scenario, since having a trace containing many **COs** over time implies to do it on an instrumented firmware. Indeed, in this case, it is not a big challenge to spot where are the **COs** since the software can perform a noisy operation to act as a trigger. This is a more difficult challenge with Screaming Channel attacks compared to traditional side channels. The same authors also explored at which frequencies the leak is detectable and exploitable [Gui+24] and concluded that the latter is present at non-harmonic frequencies and is strong enough to conduct attacks, sometimes as powerful as using harmonic frequencies. Additionally, the same authors explored if the Screaming Channel leakage may be found and exploited on more challenging devices [Gui24]. More specifically, they assessed Screaming Channels on a SoC-FPGA from the Xilinx UltraScale+ family, providing a high isolation between the digital computing part and the RF part – a direct-RF **SDR**, where the **modulation** is produced by digital hardware components without involving an analog mixer. While they found several leakage evidence, the leakage was not strong enough to mount a successful side-channel attack against AES. From the same perspective of finding a Screaming Channel leakage in another type of device, Gallagher *et al.* aimed at assessing the CC1111 and CC1310 transceivers used on satellites against Screaming Channels [Gal24a; GJ24; Gal24b]. Those studies identified a weak evidence of a Screaming Channel leakage through a *t-test*, without any successful attacks. Fanjas *et al.* [Fan+22] also introduced a new real-time triggering method called Synchronization by Frequency Detection (SFD) based on Camurati's triggering but implemented on an FPGA. Finally, Danieli *et al.* [Dan+24] tried to exploit the Screaming Channel leakage in other ways than performing cryptographic side-channel attacks. For instance, this paper exploits the direct readout of the Screaming Channels effect using various coupling sources (RAM, SPI, JTAG, NFS) to recover non-encrypted data.

*Screaming Channels extensions include automatic or real-time detection of cryptographic operations, deep-learning usage, and attacking non-cryptographic targets.*

### 3.5.2 Leakage Detection

Side-channel leakage detection is a hard problem and a whole field of research. In this section, we will present some methods to perform such leakage detection focusing on **electromagnetic radiation**.



**AUTOMATIC MODULATION CLASSIFICATION** [Automatic modulation classification \(AMC\)](#) is a problem coming from telecommunications, with the goal of identifying the modulation type and parameters given a signal. Spooner *et al.* [SBY00] introduced a method to automatically detect and classify each of a number of signal sources that can overlap in the spectral or temporal domain, by exploiting the structure of higher-order statistics of man-made [radio-frequency](#) signal. Such techniques may be leveraged in [RF](#) reverse-engineering, signal identification, and also leakage detection — since leakage can be modeled as a communication channel, but facing the challenge that the modulation scheme may be different from “man-made” signals.

**SPATIAL DETECTION** The first practical approach is to detect the leakage in space, using an motorized X-Y-Z stage, leveraging leakage generation and detection algorithm. Such an example is proposed by Werner *et al.* [Wer+18] in 2018, using a method to identify instruction-dependent leakage sources on a printed circuit board ([PCB](#)), by localizing magnetic field sources from measurements around the [PCB](#).

**FREQUENCY DETECTION** A second practical method is to generate known leakage on the target, such that it will be easier to detect and analyze them. Callan *et al.* [CZP14] proposed in 2014 to use the previously introduced micro-benchmark [ZP14] (in Section 3.3.3) about signal generation to introduce the *SAVAT* metric. This metric measures the side-channel leakage, *i.e.*, the overall signal made available to the attacker, by creating a single instruction difference in program execution. Motivated by automatizing effort, Callan *et al.* [CZP15] introduced *FASE* in 2015. This algorithm allows the uncovering of amplitude-modulated ([AM](#)) clock signals (and their harmonics) that are categorized as [electromagnetic](#) leaks, based on signal processing techniques on traces recorded using the micro-benchmarks used in *SAVAT*. Then, Wang, Callan, and Zajic [Wan+16] improved *FASE* in 2016 by automating the visual inspection of the [electromagnetic](#) spectrum. Finally, Prvulovic, Zajic, and Callan *et al.* [Prv+17] published their final improvement of *FASE* in 2017 by detecting also frequency-modulated ([FM](#)) leaks from clock signals. In Part III, we will introduce phase-modulated side channels. This work could potentially be leveraged to improve side channel leakage detection, since to our knowledge, no paper tried to analyze [phase modulation \(PM\)](#) in the time domain.

**LEAKAGE SIMULATION** A third approach is to simulate the [electromagnetic](#) leakage, instead of performing practical measurements. Buhan *et al.* [Buh+22] published a literature review about side-channel detection and prevention. However, considering [electromagnetic](#) leakage, this work concludes that it is significantly difficult to perform [electro-](#)



[magnetic](#) simulation accurately without having access to the electrical schematics of the hardware.



## Part II

### BLUESCREAM: SCREAMING CHANNELS ON BLUETOOTH LOW ENERGY

In recent years, a class of wireless devices has been demonstrated to be vulnerable to a new side-channel attack called Screaming Channels. This attack exploits distant electromagnetic side channels up to a few meters, when a coupling occurs between the digital activity and the radio transceiver of a system. This can happen in mixed-signal chips, where both digital and analog parts reside on the same silicon die. Until now, the Screaming Channel attack has mainly been demonstrated using custom firmware used in laboratory conditions or simple protocols – *e.g.*, Google Eddystone.

In this part, we evaluate an end-to-end Screaming Channel attack on a real-world firmware running on an off-the-shelf and popular Bluetooth Low Energy stack. By doing a careful analysis of Bluetooth Low Energy to find how to make the victim device leak, our results show that an attacker can manipulate the protocol such that a Screaming Channel leak happens during a radio transmission. Finally, we conducted one successful full-key recovery attack against AES using instrumented firmware and a partial-key recovery using stock firmware.



# 4

## MOTIVATIONS

**S**CREAMING CHANNELS discovery enables long-distance **electromagnetic (EM)** side-channel attacks using radio-equipment. The knowledge about its impact regarding modern **Internet of Things** protocols is currently limited, since the attack is known only since 2018. In this chapter, we will discuss why it is an important problem to evaluate the feasibility and the impact of Screaming Channels, introduce **Bluetooth Low Energy** protocol, and summarize our contribution.

This and the two following chapters are adapted from our publication at ACSAC'24 [Ayo+24a] and augmented with additional information:

Pierre Ayoub et al. "BlueScream: Screaming Channels on Bluetooth Low Energy." In: *40th Annual Computer Security Applications Conference (ACSAC '24)*. Waikiki, Honolulu, Hawaii, United States, Dec. 2024. URL: <https://hal.science/hal-04725668>

First, Section 4.1 will present our motivations for pursuing this research direction, and how impacting this attack can be. Second, Section 4.2 summarize our contributions resulting of this work, what we achieved in terms of results and performances. Third, Section 4.3 introduces **Bluetooth Low Energy** protocol and depicts some internal security features that we will attack. Finally, Section 4.4 details the differences between our work, previous work and concurrent work.

### 4.1 POTENTIAL IMPACT

Physical side channels (presented in Section 3.4) often require that the attacker is in close physical proximity with the victim target to perform his measurement. It is especially true for **electromagnetic** side-channels [QS01] that use signals recorded through **near-field (NF)** probes placed at a few millimeters of the victim device. This is limiting regarding their practical impact, mainly when considering **Internet of Things** protocols, since target connected devices can be embedded inside unreachable places. Since the countermeasures against side-channel attacks are costly — both in performance, energy, and time-consuming to deploy, if an attack is not a realistic threat and affordable for an attacker, manufacturers will not deploy any countermeasures to protect citizen's privacy.

The Screaming Channel attack (introduced in Section 3.5.1.1) discovered in 2018 by Camurati et al. [Cam+18] breaks the limitations of

*Screaming Channels may be a "game changer" for manufacturers when considering side channels countermeasures.*

*Screaming Channels broadcast side-channel leakages inside the legitimate radio transmission.*

traditional **EM** side-channels. Indeed, this paper demonstrates that **EM** side-channels can be conducted in the **far-field (FF)** region, using antennas from a few centimeters to some meters, because of a coupling phenomenon between the digital and the analog part inside a mixed-signal chip. When the transceiver of the analog part is performing a **radio transmission**, we can observe a side-channel leakage from the digital part in the spectrum of the radio transmission. This side-channel leakage first modulates the carrier signal of the radio transmission and then is amplified and broadcasted with the carrier by the radio transceiver.

A more in-depth analysis of the Screaming Channel leakage [**CFS20**] provided a better understanding of the leakage — *e.g.*, its distortions or the portability of the templates. While this analysis performs a Screaming Channel attack on Google Eddystone [**Go015**], a simple and discontinued protocol [**Blo18**], there is no prior work tackling the challenges of using Screaming Channels to target complex protocols involving multiple layers in realistic environment.

*If Screaming Channels reveals to be a practical attack on IoT protocols, the impact would be critical.*

Filling this gap is important to consider Screaming Channels attacks a realistic threat. Considering the widespread of Bluetooth Low Energy protocol (introduced in Section 4.3), being able to conduct an end-to-end attack at several meters of distance from the wireless device would be critical. Indeed, since side-channel attacks are often not taken into account in their threat model and are difficult to mitigate, billions of devices would be impacted from a distant and passive threat. Using Screaming Channels to retrieve a cryptographic key would not only allow to eavesdrop communications — but also to impersonate, control and inject traffic into existing connections.

*Our work target the key derivation mechanism of the Bluetooth Low Energy to retrieve the Long Term Key (LTK).*

In this work, we tackle the challenges raising from a complex and multi-layered protocol by attacking **Bluetooth Low Energy (BLE)**, a widespread protocol for secure wireless communications. More precisely, we attack the **BLE** key derivation mechanism – also known as the re-keying mechanism, which is a standard and fundamental security feature of the protocol. The impact of breaking this mechanism is critical since it would allow the recovery of the **long term key (LTK)** independently of the pairing method used for its generation and negotiation, including LE Secure Connections.

## 4.2 CONTRIBUTION

Our research question consist in the following:

Is the Screaming Channel attack a threat for **Bluetooth Low Energy**, despite the limitations imposed by such complex and multi-layered protocols?

**CHALLENGES** Answering this question implies to face several challenges. A specificity of the Screaming Channel leakage is that the transceiver of the target device must transmit data during the cryptographic operation, *i.e.*, emits radio waves to broadcast the leak over the air. Depending on the protocol, this period can be short and may happen independently of the cryptographic activity targeted by the side-channel attack. Moreover, side-channel attacks usually involve collecting thousands of traces, while the collection rate is limited due to protocol constraints.

*One of the main challenge is to have a radio transmission during the targeted cryptographic operation.*

**CONTRIBUTIONS** To solve those challenges, our work introduce the following contributions:

- In Section 5.4, we analyze how to manipulate the BLE protocol from an attacker perspective to find how to make the victim device “scream”, *i.e.*, execute critical cryptographic operations during a radio transmission, and how to accurately identify this timing. In addition, we developed a framework that allowed us to collect and process numerous traces over several days.
- In Section 5.5, we evaluate the Screaming Channel attack performance against the long term key (LTK) of BLE under various conditions and firmware. We demonstrate that the protocol can be exploited to fully break the AES key at the condition of reducing the radio noise of the environment during the collection process in laboratory conditions, but that this noise is problematic for an attacker in real conditions.
- In Section 6.1, we present the result of experiments using several firmware modifications, suggesting which hardware element is implicated in the Screaming Channel leakage and how it can influence its exploitation.

*Our contributions include enabling the attack on BLE, evaluating it inside several environments, and characterization of the leakage.*

Our framework is published as open-source software<sup>1</sup> and our datasets are available as open data [Ayo24].

## 4.3 BLUETOOTH LOW ENERGY

These last years, numerous Internet of Things protocols were advertised to gain market shares, such as Bluetooth Low Energy (BLE), ANT, Zig-Bee, Nb-IOT, LoRaWAN, SigFox. In this part, we studied the Bluetooth Low Energy (BLE) as a representative protocol of Internet of Things protocols.

Introduced in 1998, Bluetooth [Gro] is today the most widely used protocol to exchange data between mobile devices in short ranges.

<sup>1</sup> Code: [https://github.com/pierreay/screeaming\\_channels\\_ble](https://github.com/pierreay/screeaming_channels_ble)

Mainly implemented into [Internet of Things](#) devices, smart devices such as phones or watches, human-machine interfaces such as keyboards, the market shares encompass 5 billion of Bluetooth devices and is expected to double in less than 10 years [[Lar24](#)]. It works in the 2.4 GHz band, itself split between 80 channels of around 1 MHz bandwidth, where each channel is using the [Gaussian frequency-shift keying](#) modulation. There is two forms of Bluetooth:

**BLUETOOTH CLASSIC** The first developed form of Bluetooth, including the *Basic Rate (BR)* and the *Enhanced Data Rate (EDR)*, often abbreviated as *Bluetooth BR/EDR*.

**BLUETOOTH LOW ENERGY** The second form was introduced in the 4.0 specification, often abbreviated as *Bluetooth LE* or *BLE*. It focuses on low-power consumption for shorter ranges — *e.g.*, below 10 meters.

In this section, we focus on the security and the radio transmission scheme of the Bluetooth Low Energy<sup>2</sup>, as we will exploit those mechanisms.

#### 4.3.0.1 Security

The Bluetooth specification describes numerous security mechanisms. In this section, we will focus on how encryption keys are generated and used to encrypt a communication channel.

**PAIRING** The pairing procedure is used to exchange a set of keys (including an encryption key) that are generally stored for future use. The stored encryption key is known as the [long term key \(LTK\)](#). The specification defines two pairing methods [[Gro](#), p. 1626]:

*Pairing allows exchanging a set of keys that will be used to secure the communication link.*

- “LE Legacy Pairing” method will negotiate a common key ([short-term key \(STK\)](#)) between the two devices to establish an encrypted session. This encrypted session will be used to exchange a set of keys, including the [long term key \(LTK\)](#).
- “LE Secure Connections” method uses the Elliptic Curve Diffie-Hellman (ECDH) algorithm to agree on a shared [LTK](#). In this case, the [LTK](#) is directly used to establish the encrypted session to exchange the other keys securely.

At the end, the [LTK](#) is shared by the two devices, being the key that an attacker will want to retrieve. Our attack focus on a later stage of the protocol, and our approach is independent of the pairing method in use.

<sup>2</sup> The reader may refer to the “Getting Started with Bluetooth Low Energy” book from [Townsend et al.](#) [[TCA14](#)] for an introduction to this protocol.



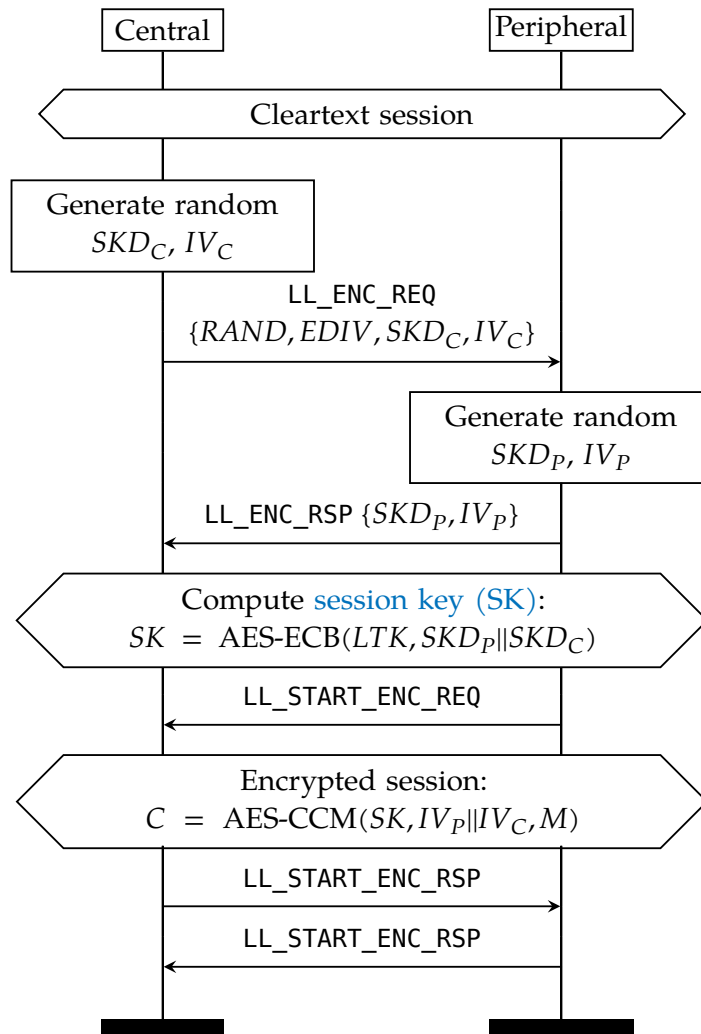


Figure 4.1: Session key derivation in BLE. The ciphertext  $C$  will be sent over the air to transmit the message  $M$ .

**SESSION KEY DERIVATION** The pairing procedure is only needed the first time the two devices are used together. For each session (connection of the devices), a new session key will be generated using the **long term key (LTK)** as a master key during a **key derivation** operation. Following the pairing, both devices will also keep two values, **RAND** and **EDIV**, used to retrieve the **LTK** from the security database storing all **LTK** from the previous pairing. The two paired devices can now establish a secure connection with link-layer encryption by computing a common **session key (SK)** if they share a common **LTK** [Gro, p. 2957]. As depicted in Figure 4.1, the **session key (SK)** is derived by performing an **AES** in **ECB** mode using the **LTK** as the key and a public random nonce as the plaintext. This random nonce, also known as the **session key diversifier (SKD)**, is composed of two concatenated random numbers respectively generated by the Central ( $SKD_C$ ) and the Peripheral

*After a pairing, two devices will use the generated keys to secure and encrypt the communication link.*

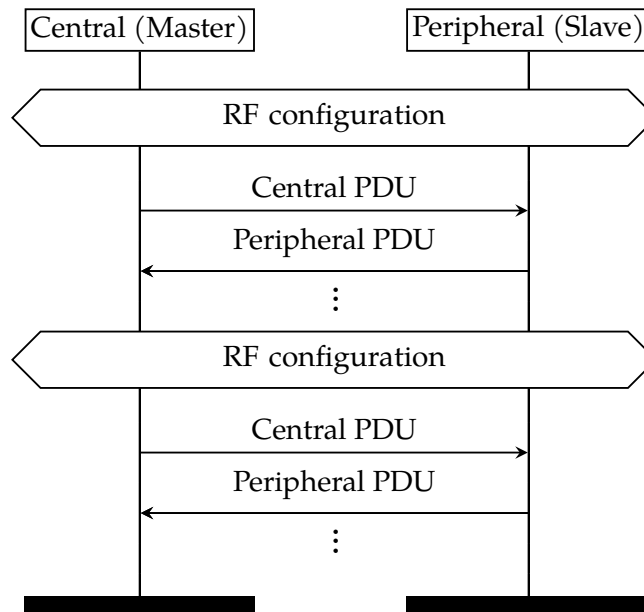


Figure 4.2: Standard BLE communication divided in CE according to Hop Interval  $H$ .

( $SKD_P$ ).  $SKD_C$  and  $SKD_P$  are transmitted over-the-air in plaintext using the control **protocol data units (PDUs)** `LL_ENC_REQ` and `LL_ENC_RSP`. After the **SK** computation, the two devices perform a three-way handshake to start the encrypted session by sending empty control **PDUs**. The Peripheral sends an unencrypted `LL_START_ENC_REQ`, the Central and the Peripheral responds with an encrypted `LL_START_ENC_RSP` and `LL_START_ENC_RSP`, respectively.

#### 4.3.0.2 Radio transmission

**TRANSMISSION TIMING** A standard **BLE** communication [Gro] is divided into **connection events (CEs)** following the **time-division duplex** mechanism (as illustrated in Fig. 4.2). During a **connection event (CE)**, the node initiating the connection called the Central (formerly called “Master”), sends at least one packet to the second node involved in the connection, the Peripheral (formerly called “Slave”). After a short inter-frame space of  $150\mu\text{s}$ , the Peripheral sends the response packet. For synchronization purposes, this simple communication pattern is repeated for each **CE**, whether data needs to be transmitted or not – e.g., packets with an empty data field are therefore often transmitted. A field called **More Data (MD)** bit can be set in the **protocol data unit** to indicate that the **CE** should not end after the current **PDU** because a device (Central or Peripheral) has more data to transfer during the current **connection event**. If the **More Data (MD)** bit is set to 0, no further transmissions will occur during the **CE**. Otherwise, the devices

*Two devices, a Central and a Peripheral, exchanges packets synchronized in time.*

repeat the communication pattern until the MD bit is not set anymore or until the CE's time window is exhausted.

**TRANSMISSION FREQUENCY** Between each CE, the transmission frequency is changed according to the channel hopping algorithm. The radio of both the Central and the Peripheral are then turned off to be re-configured for another frequency. The duration of a CE is equal to  $\text{connInterval} = H * 1250\mu\text{s}$ , with the hop interval  $H \in [6; 3200]$ . The next frequency, *i.e.*, the next channel, is chosen between all standard channels defined from the specification. The channel selection can be influenced by the channel map, a 5-byte value where every bit indicates if a channel is blacklisted or not. The Central transmits the initial channel map during the connection initiation but can be modified at any time during the connection, allowing a fast adaptation to changes in the radio environment (*e.g.*, interference).

*The channel frequency is hopping after a terminated transmission.*

**FREQUENCY HOPPING** A Bluetooth communication (excluding advertisement) can be located between 2.404GHz and 2.478GHz on the spectrum due to the frequency hopping (FH) algorithm. Intercepting communication from such a protocol can be challenging because consumer-grade radio equipment cannot monitor a large enough band in the spectrum. When the channel hopping sequence can be neither modified nor known *a priori* by an attacker, a possible solution is to rely on SDR flexibility and FPGA-based acceleration in order to decode a wideband region (*e.g.*, 200 MHz) in real-time [Lav22].

*Frequency hopping is a challenge for sniffing Bluetooth Low Energy at the radio layer.*

#### 4.3.0.3 Implementations

The specification defines 2 layers that communicates with each other, known as the Host-Controller Interface (HCI):

**HOST** Implements the layer corresponding to the transport protocol which is interoperable between Controllers.

**CONTROLLER** Implements the Link Layer (known as *LE LL*), which is the low-level and real-time protocol which communicate with the radio hardware to schedule packets reception and transmission.

Several implementations of the Bluetooth Low Energy exists on commercial products. In the following, we introduce the major stack implementations for embedded systems that are relevant to our work:

**NORDIC SOFTDEVICE** [SEM24] Proprietary and closed-source wireless stacks for nRF51, nRF52 and nRF53-series devices from Nordic Semiconductor, including the BLE stack. Its closed-source nature makes research difficult because we cannot instrument it or use a custom software encryption.

ZYPHYR OS [PR024] Free and open-source **real-time operating system** designed for embedded systems, supporting numerous architectures, including ARM-based cores, available under the Apache license. It includes a complete **BLE** stack — including a Host and a Controller for nRF52 and nRF53-series — as well as a **BR/EDR** Host.

APACHE NIMBLE [FOU24B] Free and open-source **BLE** stack developed by Apache for the Mynewt **real-time operating system** [Fou24a] available under the Apache license. The Host implementation supports as many chipsets based on ARM-based cores as Mynewt supports, while the Controller implementation mainly supports the nRF51 and nRF52 chipsets from Nordic Semiconductor.

In this part, we focused on the Apache NimBLE stack because its modularity allowed us to modify it as our needs, and it has a stable support for the nRF52 — which was extensively studied in previous research [Cam+18; CFS20].

## 4.4 RELATED WORK

In this section, we will discuss and compare our work with two important work related to our research question:

- *Camurati et al.* [CFS20], which evaluates the Screaming Channel attack on *Google Eddystone* in 2020.
- *Cao et al.* [Cao+23], which performed a side-channel attack on the **Bluetooth Low Energy (BLE)** key derivation mechanism in 2023.

### 4.4.1 Prior Work: Attack on Google's Eddystone

*While Screaming Channels have been already evaluated against the Eddystone protocol, no work has been done for assessing its threat against complex protocols inside realistic environments.*

**PRELIMINARY ASSESSMENT** A preliminary assessment of the risk on modern protocols induced by the Screaming Channels attack has been conducted by *Camurati et al.* [CFS20] by exploiting Google's Eddystone protocol. Eddystone [Goo15] is an application-level protocol on top of **BLE** used for beacons, *i.e.*, small IoT objects broadcasting information (*e.g.*, a URL). While the *Camurati et al.* [CFS20] work is the first one to assess Screaming Channels on a wireless protocol, this work also suffers from serious limitations. First, it only evaluates the attack using a synthetic setup with a **radio-frequency (RF)** coaxial cable, drastically limiting the noise of the leakage. Second, Eddystone is an overlay to **BLE**, using it as a transmission layer, which is not representative of wireless protocols since:

- It is a non-standard, not widespread and discontinued [Blo18] protocol specific to the beacon use case.

- The security of the protocol is implemented at the application level using Bluetooth services. It implies that the **cryptographic operation** is triggered using a *Characteristic Read*, an operation built over the application layer of **BLE** that does not involve the native security mechanisms of **BLE**, facilitating the exploitation.

**COMPARISON WITH OUR WORK** In contrast, in this work, we aim to overcome those limitations by using a realistic setup for our evaluation and performing an end-to-end attack:

- Regarding the attacker receiver, we use an antenna at different distances from the target instead of an RF coaxial cable, which is a necessary requirement to claim that an attack is a realistic threat.
- Regarding the attack target, we attack the standard key derivation mechanism of the **BLE** protocol instead of the Eddystone application-level implementation, which is an important challenge since the cryptographic operation trigger is not directly under the attacker control.
- Finally, regarding the victim firmware, we attack a standard open-source **BLE** stack, which do not allow using post-processing averaging techniques, for increasing the **signal-to-noise ratio (SNR)** of our traces.

*In our work, we attack the fundamental key derivation mechanism of BLE inside realistic environments.*

In summary, the differences lie into the layer implementing the cryptographic operation, implying more or less control for the attacker, and in the **electromagnetic** environment used for the evaluation.

#### 4.4.2 Parallel Work: Side-channel attack on Bluetooth

A necessary requirement of performing a Screaming Channel attack on a cryptographic target is to be able to perform a conventional side-channel attack on such cryptographic algorithm. The key derivation mechanism of the Bluetooth Low Energy is leveraging the AES (introduced in Section 2.3) symmetric cryptographic algorithm. As several other algorithms from this category, this algorithm is vulnerable to side-channel attacks if the attacker is able to know the plaintext. Assessing the vulnerability to side-channel attacks of the **Bluetooth Low Energy (BLE)** requires an analysis on how the specification use AES and how it handles the different variables. This work has never been done at the date of the beginning our work in 2022. While we planned to add this analysis into our publication, concurrently to our work, Cao *et al.* [Cao+23] analyzed the session key derivation mechanism (also known as re-keying) of the **BLE** protocol to assess its side-channel

*Assessing the BLE vulnerability to side-channel attack has never been done before our work.*

*The same year as our work, another paper analyzed the BLE re-keying mechanism vulnerability to side-channel attacks.*

resistance.<sup>3</sup> *Cao et al.* show a conventional **electromagnetic (EM)** side-channel attack, with an **EM** probe in the vicinity of the microcontroller, at low frequency, without the radio activated. They compare the performance of traditional correlation attacks against deep learning-based attacks and do not address any of the challenges of a remote attack from the radio channel — in particular, the challenges of synchronization, triggering, and having a radio transmission during the cryptographic operation.

*This concurrent work perform the same side-channel as our work concerning the Bluetooth Low Energy.*

**SIMILARITIES WITH OUR WORK** The work from *Cao et al.* [Cao+23] shares some similarities with our work:

- They performed the same analysis and conclusion as us concerning the **BLE** vulnerability to side-channel attacks. Their side-channel strategy are similar — regarding the **LTK**, the **session key diversifier (SKD)**, the **BLE** requests to sniff, as well as attacking the *SubBytes* of the 1st round of AES.
- They used NimBLE, an open-source **BLE** stack, as the target firmware running the TinyCrypt software implementation of AES.
- Their threat model — except about attacker distance, specific to Screaming Channel — and the attack scenarios are similar.

*However, we tackle challenges inherent to a realistic threat for IoT protocols.*

**DIFFERENCES WITH OUR WORK** However, it also differentiate for our work in several ways:

- They performed the side-channel attack using conventional **EM** traces acquired using a **near-field** probe, while we use **far-field** traces acquired using an antenna.
- They use traditional and expensive side-channel attack hardware, consisting in oscilloscope and probes, while we use cheaper **software-defined radio** and antennas.
- They used a trigger signal generated from the target device using a **GPIO**, which is only possible in a synthetic setup. Instead, we use the open-source framework WHAD [CC24], which we use to: (1) trigger the **cryptographic operation** from the target device, (2) generate a software-based trigger signal to our **software-defined radio** instrumentation software. This allows us to make the attack a lot more realistic.

In summary, we observe that none of these papers evaluated the Screaming Channels attack on a complex multi-layered protocol such as **BLE** in a realistic scenario.

<sup>3</sup> A reader deeply motivated to understand how to perform a side-channel attack on the Bluetooth Low Energy is encouraged to read the great work from *Cao et al.* [Cao+23] published as the SKEBLE attack.

# 5

## ATTACKING BLUETOOTH LOW ENERGY

**B**LUETOOTH LOW ENERGY is an interesting target to attack with Screaming Channels, because it is a widespread and mature protocol. Moreover, the frequency hopping mechanism is representative of other [Internet of Things](#) protocols. In this chapter, we present how the Screaming Channels attack can be engaged in the [BLE](#) context.

First, [Section 5.1](#) and [Section 5.2](#) will present the threat model and the complete attack sequence. Second, [Section 5.3](#) summarize the different experimental setups we used during our work, and for which purpose. Third, [Section 5.4](#) details [BLE](#) protocol manipulation from the attacker to enable the Screaming Channel attack. Last, [Section 5.5](#) exploits and evaluate the Screaming Channel attack on [BLE](#).

### 5.1 THREAT MODEL

We consider an attacker that aims to obtain the [LTK](#) used to generate key material for secure communications between two paired devices: a victim Peripheral and a victim Central. The attacker does not need to monitor the pairing phase, and the attack is independent of the pairing method used (LE Legacy Pairing or LE Secure Connections). The attacker is able to sniff and inject packets at the physical layer. More precisely, the attacker will need to:

- *Sniff* a [BLE](#) encryption initiation (LL\_ENC\_REQ) packet between the Central and the Peripheral, to obtain the parameters sent in the clear from the Central (addresses, RAND and EDIV),
- *Interrupt* a previous connection. While not strictly required, this makes the attack faster by forcing the Central and Peripheral to establish a new connection and collect parameters, instead of waiting for a new connection.
- *Impersonate* the victim Central (using the above parameters), which was previously paired with the victim Peripheral,
- *Initiate* a connection and interact with the Peripheral victim through the [BLE](#) protocol,
- *Repeat* session key establishment on the Peripheral side using this connection, which triggers an encryption using the [LTK](#) as a key,

*Using a standard IoT security framework and an SDR, the attacker is able to perform sufficient interactions with the victim to enable the attack.*



- *Record* a radio signal over the air during the encryption, at the physical layer, (e.g., using an antenna connected to a [software-defined radio \(SDR\)](#)).

In our attack, we used profiled algorithms, where an attacker can build a template to learn the leakage model using a similar device that can be instrumented prior to the attack. Note that this is not strictly required when exploiting screaming channels because non-profiled attacks may also be performed. Once the attack is successful ([LTK](#) is recovered), the attacker can decrypt data sent over the air during [BLE](#) communications between the two victim devices – for previous and future connections [[Ant23](#)], provided that the packet traces can be collected. According to the state of the art in [BLE](#) security, these capabilities are realistic and can be performed with affordable open-source tools [[Wu+24](#)].

## 5.2 ATTACK OVERVIEW

Figure [5.1](#) presents a high-level overview of our attack strategy, composed of three different phases.

**1<sup>st</sup> PHASE (OFFLINE)** In the first offline phase, steps may occur at different locations and times before the attack:

1. *Template creation*: The attacker prepares a template using a different device but a similar model to the victim Peripheral [[CFS20](#)]. Under controlled conditions, he collects a high number of traces corresponding to the session key derivation, where an AES encryption with a controlled random input ([LTK](#), [SKD](#)) is performed by the victim Peripheral (see Fig. [4.1](#) from Section [4.3.0.1](#)). This is specific to the profiled side-channel attacks we used during this work, and this is not a strict requirement for the attack.
2. *Legitimate pairing*: The victim Peripheral and victim Central perform a pairing in secure conditions, agreeing on a shared [long term key \(LTK\)](#) and its associated parameters (RAND and EDIV). The attacker is not present during this phase.

*The attacker needs to listen to a legitimate connection to sniff its plaintext parameters.*

**2<sup>nd</sup> PHASE (ONLINE)** In the second online phase, the attacker needs to be active and in the victim's proximity during the following steps:

3. *Legitimate connection sniffing*: After the pairing phase, the attacker passively observes one legitimate connection and encrypted session establishment between the victim Central and the victim Peripheral. From the `CONNECT` message, he collects the victim's Central Bluetooth address (`BD_ADDR`) and the address type – used



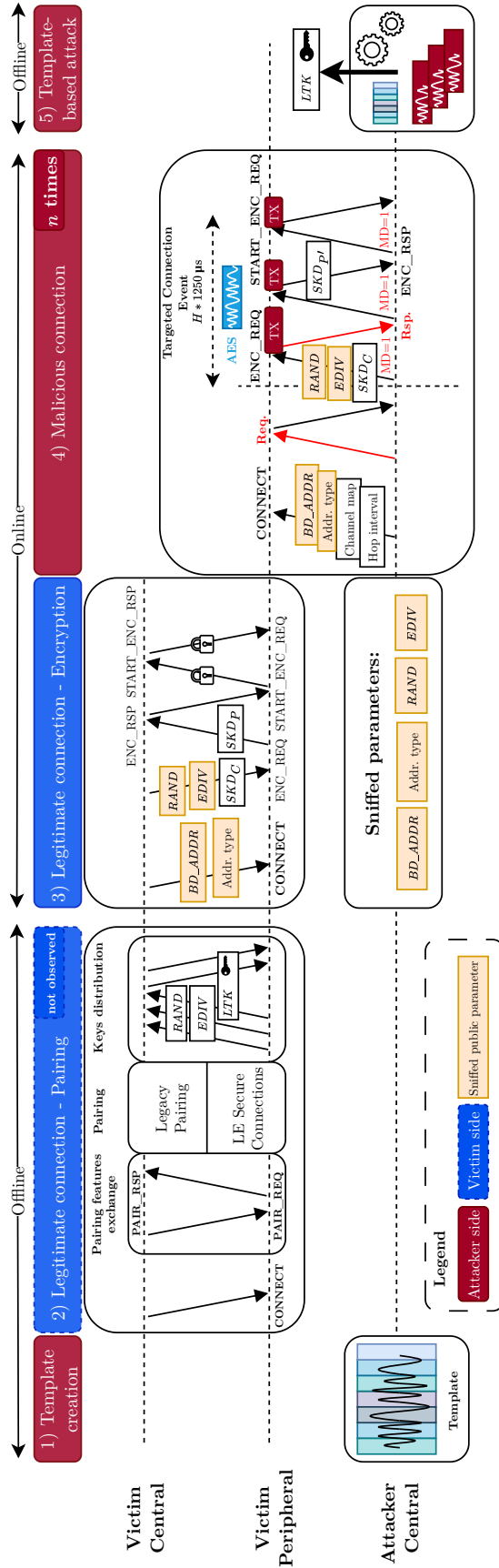


Figure 5.1: Attack overview.

*The attacker will repeat a high number of partial connection attempts to collect side-channel traces.*

later on to impersonate the victim Central. From the LL\_ENC\_REQ, he collects the RAND and EDIV associated with the LTK in use – used later on to trigger a similar session key derivation on the victim Peripheral without knowledge of the LTK.

4. *Malicious connections*: The attacker repeatedly performs  $n$  connections by:
  - a) *Impersonating* the victim Central,
  - b) *Injecting* controlled channel map and hop interval,
  - c) *Starting* a BLE procedure requiring a response to generate an interleaved procedure,
  - d) *Triggering* the session key derivation by initiating the encrypted session establishment,
  - e) *Setting* the MD bit of every transmitted PDU to 1.

This protocol manipulation maximizes the victim Peripheral **radio transmission** duration, increasing the probability that the session key derivation occurs concurrently with the **radio transmission (TX)** period, therefore generating a Screaming Channel leakage. Without this protocol manipulation and using only a passive attacker device, the probability of having a Screaming Channels leakage would be weak or close to zero. Simultaneously, the attacker records the **electromagnetic radiation** at the correct frequency, leveraging the knowledge of the channel map in use. We present the technical details of this protocol manipulation in Section 5.4.

*Finally, the attacker leverage the traces to perform the side-channel attack.*

**3<sup>rd</sup> PHASE (OFFLINE)** Lastly, once enough traces have been collected, the third offline phase consists of:

5. *Template-based attack*: The attacker performs a template-based side-channel attack to recover the LTK from the collected traces using the template previously created. If needed, the attacker can perform a key enumeration to brute-force the remaining bits incorrectly recovered by the side channel. This is possible because the attacker can capture encrypted but predictable traffic, which can be used as an oracle during key enumeration. If the attack is successful, the full LTK is retrieved.

### 5.3 EXPERIMENTAL SETUP

Fig. 5.2 illustrates our hardware and software experimental setup in laboratory conditions (without the Central Victim described in Section 5.2).

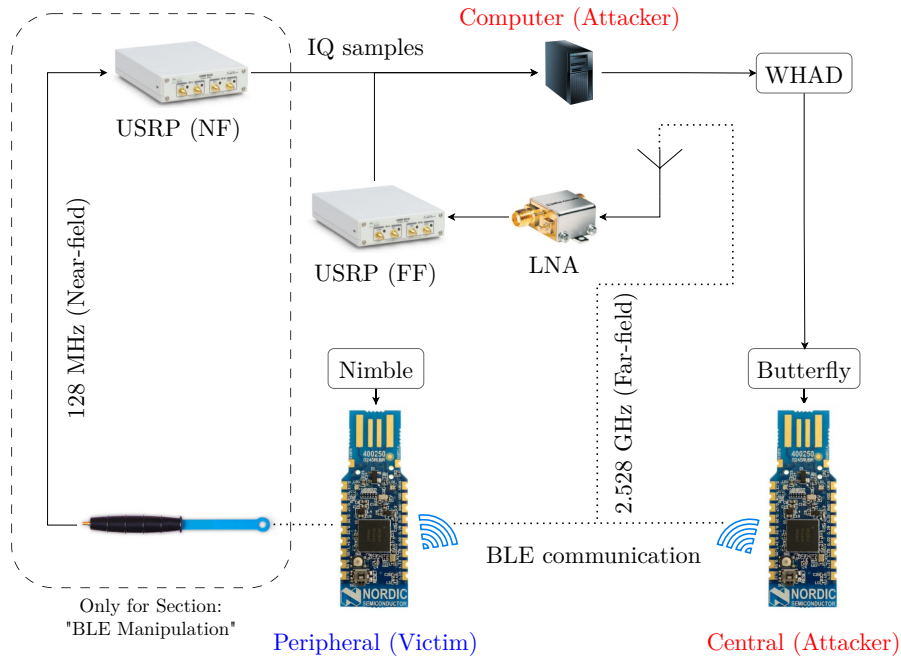


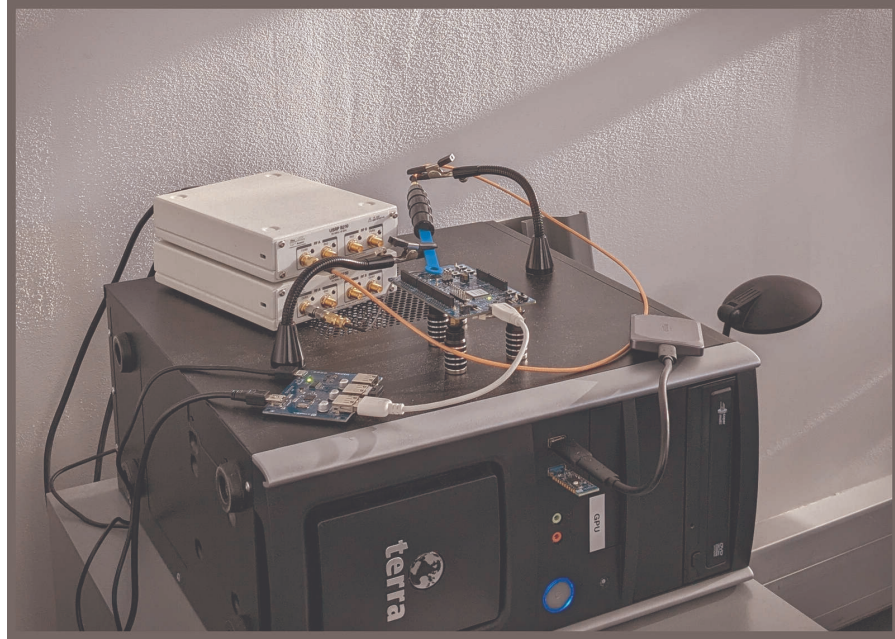
Figure 5.2: The experimental setup.

**HARDWARE** We use two **software-defined radios** capable of recording up to 56 MHz of bandwidth (USRP B210 [Res]). The USRP NF, only used in Section 5.4, is connected to a TekBox TBPS01 **near-field (NF)** probe [Tek24a] placed at 1 cm of the target SoC and tuned at the 2nd harmonic of the CPU clock ( $f_{nf} = 2 * f_{clock}$ ). The USRP FF is connected to an antenna placed in the **far-field (FF)** (typically, more than 30 cm) from the target and tuned at the carrier frequency added to the previous frequency  $f_{ff} = f_{carrier} + f_{nf}$ . Typically,  $f_{clock} = 64$  MHz for the CPU clock of the nRF52832 and  $f_{carrier} = 2.420$  GHz for the Bluetooth channel 8. Depending on the experiment, we used an omnidirectional [GoT24] or a directional [TP-24] antenna to increase the leakage gain. A LNA from Mini-Circuits [Min24] is plugged between the antenna and the SDR, increasing the SNR of our recorded signal. The two SDRs are connected to a standard desktop computer through USB 3.0, sending raw I/Q during the recording. The attacker's dongle is a Nordic Semiconductor nRF52840 dongle, while the victim's device is a Nordic Semiconductor nRF52832 development kit (PCA10040), a well-known target for Screaming Channels analysis.

To characterize the impact of the protocol manipulation on the leakage in Section 5.4, the full setup, including the USRP used for **near-field (NF)** measurements, was used. Fig. 5.3 is a picture of this setup. We observe that the attacker dongle plugged into the front face of the computer, as well as an external storage device for the large datasets. The target board is powered *via* an USB switch, and its **RF** output is redirected into a coaxial cable connected to the USRP FF. Moreover, the **near-field (NF)** probe is placed above the external C6 and C7 capaci-

*An antenna and a SDR are the basic needs of this attack.*

*A first setup using two radios has been used to characterize the protocol manipulation on the leakage.*



**Figure 5.3:** Leakage characterization setup using two synchronized USRPs used in Section 5.4.

**Table 5.1:** Firmware used during this work.

Name	Code	Encryption nb.	AES inputs	Radio packets
$F_{\text{custom}}$	Custom C	Arbitrary	Controlled	Dummy
$F_{\text{instru}}$	NimBLE	Arbitrary	Controlled	Real BLE
$F_{\text{default}}$	NimBLE	1	Known only	Real BLE

tors of the nRF52 and connected to the USRP NF. The two USRPs are synchronized using a software-based solution, re-aligning traces using timestamps and correlations.

To evaluate the attack in a realistic scenario and its performance in Section 5.5, the setup excluding the USRP NF was used (*i.e.*, only including the USRP used for far-field (FF) measurements). Fig. 5.4 is a picture of this setup. Compared to the previous setup, we removed one USRP but added a laboratory power supply to power up a low-noise amplifier (LNA) added between the USRP and the antenna. The antenna is placed at 1.2 meters from the target board.

*A second setup using a single radio has been used to evaluate the attack in challenging conditions.*

**SOFTWARE** The computer runs our custom Python instrumentation library built on the WHAD [CC24] framework, which controls the attacker’s dongle. The attacker’s dongle (nRF52840) is running our modified version of ButterFLy [Cay24], a BLE firmware initially developed for the InjectaBLE attack [Cay+21a] allowing to accurately inject link-layer traffic.



Figure 5.4: Collection setup using the directional antenna.

**FIRMWARE VARIATIONS** During our evaluations, we used 3 firmware for the victim target depending on our needs presented in Table 5.1:

- $F_{\text{custom}}$  is a custom firmware built upon homemade C code based on Camurati et al. [Cam+18] work for experimental purposes<sup>1</sup>.
- $F_{\text{instru}}$  is an instrumented firmware built upon the NimBLE [Fou24b] stack for evaluating the attack in favorable conditions to the attacker.
- $F_{\text{default}}$  is a stock firmware built upon the NimBLE [Fou24b] stack for evaluating the attack in realistic conditions.

*Using different firmware allowed us to independently evaluate the impact of controlled parameters.*

$F_{\text{instru}}$  and  $F_{\text{default}}$  running, respectively, a modified and original version of the Peripheral example firmware from NimBLE [Fou24b]<sup>2</sup> — introduced in 4.3.0.3. This BLE stack provides AES through either a hardware implementation if the SoC implements it, or a software implementation using the TinyCrypt [Int24] library otherwise. In our evaluation, we use the software AES since its leakage is stronger than that of the hardware AES, and the focus of the paper is not to assess the difficulty of attacking hardware-based implementations. Let us highlight that, to the best of our knowledge, no prior work managed to perform a successful Screaming Channels attack on a cipher in hardware. All radio transmissions use the GFSK modulation scheme whenever the packets are dummy or real BLE.

<sup>1</sup> The code is located at: [https://github.com/pierreay/screaming\\_channels\\_poc](https://github.com/pierreay/screaming_channels_poc)

<sup>2</sup> The code is located at: [https://github.com/pierreay/screaming\\_channels\\_nimble](https://github.com/pierreay/screaming_channels_nimble)

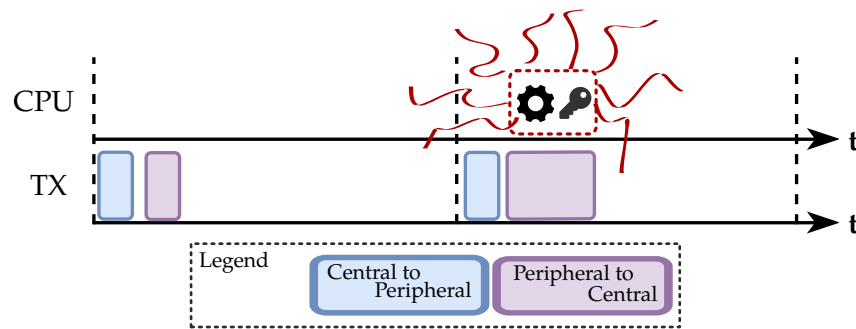


Figure 5.5: One of the challenge of Screaming Channels is that a radio transmission must happen at the same time of a sensitive operation to have a [compromising emanation](#).

## 5.4 BLUETOOTH LOW ENERGY MANIPULATION

### 5.4.1 Challenges

Three fundamental requirements of the Screaming Channel attack become challenging when using a complex protocol instead of custom firmware.

*While these challenges involves engineering efforts, answering about their feasibility would provide insights to address the scientific question about far-distance side-channel threats.*

**CHALLENGE 1: TRANSMITTING RADIO SIGNALS WHILE THE CRYPTOGRAPHIC OPERATION IS PERFORMED** The victim must transmit radio signals while the [cryptographic operation](#) is performed, as illustrated in Fig. 5.5. Otherwise, the radio transmission is off, and no data is leaked over the radio. However, both the encryption and the transmission are short (in the order of 100  $\mu$ s) and may happen in sequence, one after the other. Indeed, the transmission is a power-consuming operation and will be made as short as possible. Relying on the chance that both operations occur simultaneously significantly reduces the probability of success of the attack and adds a significant time overhead.

**CHALLENGE 2: RECORDING AT THE CORRECT TIME** Second, the attacker has to collect data at the right time, as the victim is leaking information only for a short duration. Contrarily to a custom firmware designed to highlight the effect itself [[Cam+18](#)], on a real protocol, the attacker cannot artificially activate the [radio transmission](#), repeat the [cryptographic operation](#) or use an artificial triggering mechanism (like a GPIO pin indicating that the encryption started).

**CHALLENGE 3: RECORDING AT THE CORRECT FREQUENCY** Third, the attacker has to record the signal at the right frequency, as the frequency of the physical channel used by the protocol may not be constant. This is rendered difficult because (1) the leakage can be spread over numerous



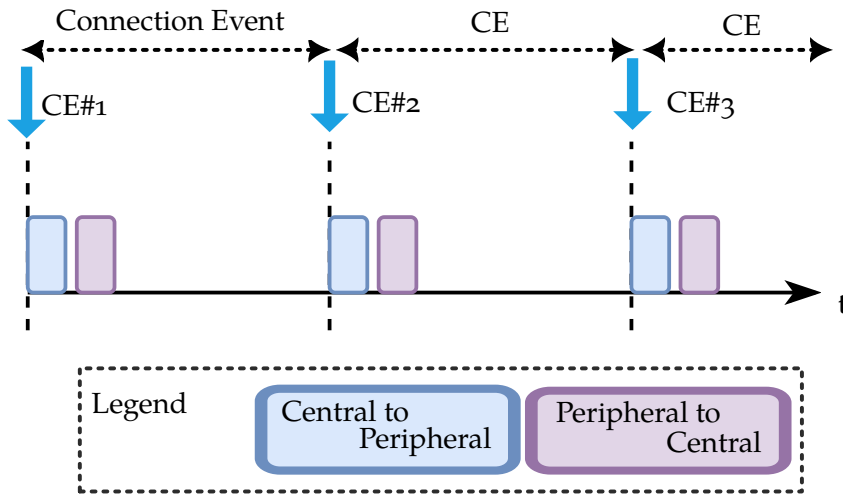


Figure 5.6: A BLE connection is divided into successive connection event (CE), that the attacker can use to target a specific moment in time.

frequencies (2) the leakage can be impacted by interfering transmitters (3) the rapid frequency hopping of BLE.

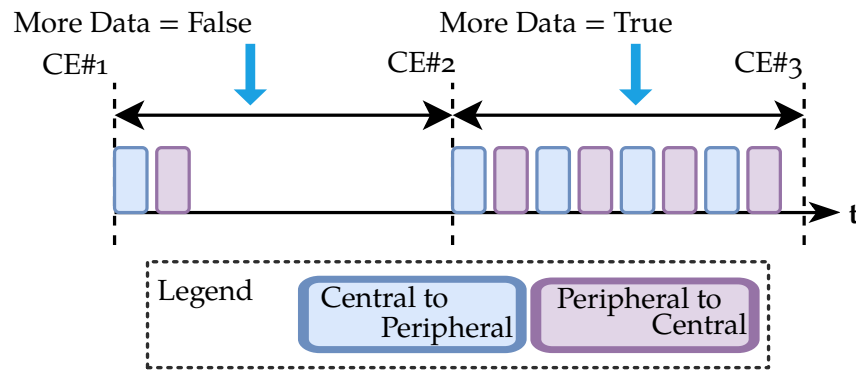
**SOLUTIONS OVERVIEW** Therefore, we developed a framework to solve those challenges by leveraging the ButteRFly firmware on the attack device presented in Section 5.3. This attack device can control low-level parameters of the Bluetooth communication, and, while staying fully compliant with the BLE protocol, allows us to finely control the victim device behavior. Using those low-level messages, we can indirectly force the victim to increase the duration of the TX, making it transmit radio messages while performing encryption and expose a Screaming Channel leakage.

#### 5.4.2 Methodology

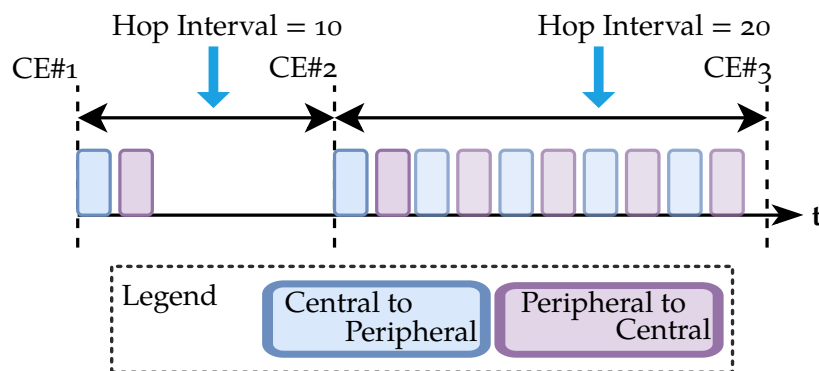
In this section, we will address the different challenges presented in Section 5.4.1 and expose our methodology.

**CHALLENGE 1: TRANSMITTING RADIO SIGNALS WHILE THE CRYPTOGRAPHIC OPERATION IS PERFORMED** According to the Bluetooth specification [Gro], we listed a set of parameters that may influence the TX timing or increase the TX duration. For comparison purposes, we measure that the AES took approximately 250  $\mu\text{s}$  to fully execute on our target board – using AES-ECB from TinyCrypt provided by Apache Mynewt. However, the *SubBytes* operation – which is the side-channel target – took only 10  $\mu\text{s}$  to execute. We used the following parameters:

*Manipulation of BLE parameters by the attacker allows making the victim transmitting during the cryptographic operation.*



**Figure 5.7:** The attacker can increase the number of packets inside one **connection event (CE)** by set the **More Data (MD)** bit to 1. This increase the probability of having a transmission during a sensitive operation.



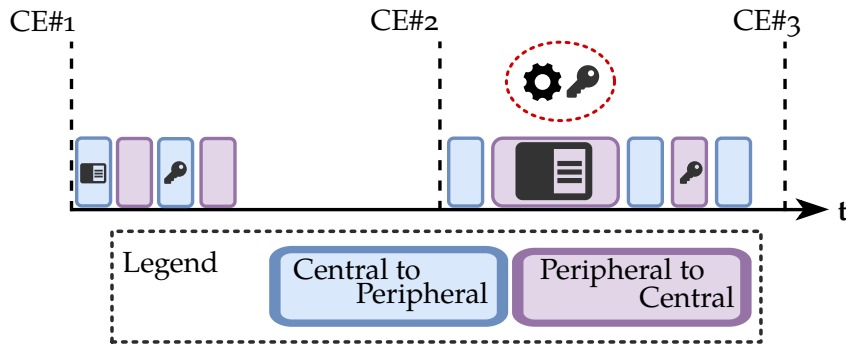
**Figure 5.8:** The attacker can increase the duration of a **connection event (CE)** by increasing the Hop Interval value. This decrease the number of radio reconfiguration for a specific time window, thus increasing the probability of having a transmission during a sensitive operation. If the **More Data (MD)** bit is set to 1, even more packets can be transmitted during a **CE** (gray shaded).

**CONNECTION EVENT** Defining a precise **CE** at which the attacker will send the **LL\_ENC\_REQ PDU** allows influencing the **cryptographic operation** execution in time. This is illustrated by Fig. 5.6.

**MORE DATA BIT** We used this bit indicating that more data needs to be sent to increase the number of exchanges between the attacker Central and the victim Peripheral during a single **CE**, hence, without a radio reconfiguration. This bit alone only allows us to send more data, and it does not send the data in itself. We set the **MD** bit when sending our **LL\_ENC\_REQ PDU** while staying compliant with the specification. This is illustrated by Fig. 5.7.

**HOP INTERVAL** Increasing this parameter allows increasing the **CEs** duration. With the **MD** bit set to 0, this parameter modifies the interval between two subsequent **TX**, *i.e.*, one **radio reception**





**Figure 5.9:** By sending multiple requests concurrently (*i.e.*, interleaving procedures), the attacker can make the victim transmit an answer during a sensitive operation. The larger will be the answer by the victim device, the higher will be the probability to have a radio transmission during a sensitive operation.

(RX) slot and one TX slot from a Peripheral perspective. On the contrary, with the MD bit set to 1, this parameter increases the number of RX and TX cycles that can fit inside the window of one CE, which is desirable to prevent the radio reconfiguration during the **cryptographic operation**. We set the Hop Interval when sending the CONNECT\_IND PDU compliant with the specification. This is illustrated by Fig. 5.8.

In the terminology used by the Bluetooth Core Specification, a Procedure is a sequence of packets, including requests and responses. We use the term “interleaved procedures” when sending multiple procedures simultaneously, resulting in interleaved request-response patterns. Having the MD bit set to 1, we sent interleaved procedures to increase the TX time during the system activity of the victim Peripheral. Before sending the request that will trigger the **cryptographic operation**, we sent other dummy requests that will force the Peripheral to send responses back. The choice of the dummy request can be important because some requests require larger responses than others, increasing the TX duration. The goal of this strategy (illustrated in Fig. 5.9) is to increase the TX time of the Peripheral during its system activity, including the **cryptographic operation**.

**CHALLENGE 2: RECORDING AT THE CORRECT TIME** The **cryptographic operation** occurs between two specific PDU [Gro, p.2843, p. 2845]: the LL\_ENC\_REQ PDU sent by the Central and the LL\_START\_ENC\_REQ PDU sent by the Peripheral. LL\_ENC\_REQ identifies the paired devices and provides the Central random value ( $SKD_C$ ), required to compute the **session key**. On the other hand, the LL\_START\_ENC\_REQ needs the **session key** as the link is encrypted [Gro, p.2843, p. 2845]. Our attack firmware can monitor the connection, report received packets, and re-

*The attacker can also send multiple procedures to make the victim transmitting.*

*Leveraging our sniffer, we are able to know in which temporal window the cryptographic operation occurred.*

port parameter values like the current **CE** number and conditionally execute a function with low latency. We leveraged those features to send the **LL\_ENC\_REQ PDU** at a chosen **CE**, start the recording at another specified **CE**, and stop it when the **LL\_START\_ENC\_REQ PDU** has been received. This guarantees the recording of potential Screaming Channel leakage during the **session key** derivation.

*Leveraging a protocol feature, we are able to force a known set of frequency to be used.*

**CHALLENGE 3: RECORDING AT THE CORRECT FREQUENCY** **BLE** uses frequency hopping, which makes it hard to record at a frequency corresponding to the leakage. The protocol allows for a Central to specify a channel map inside **ChM** field of the **CONNECT\_IND PDU** [Gro, p. 2688] sent during the connection establishment. By setting the channel map to  $0x300$ , we can force the Peripheral to only use channels 8 and 9, corresponding to frequencies 2.420 GHz and 2.422 GHz. This technique was already used in previous work targeting Google’s Eddystone protocol [CFS20]. By recording at  $f_{\text{carrier}} + 2 * f_{\text{carrier}} = 2.548$  GHz using at least 4 MHz of bandwidth, we are now sure to match the Screaming Channel leakage frequency allowing us to capture both channels in the same record. However, as seen in Section 6.1.2, this frequency is not the only one where the leak can be observed and exploited.

### 5.4.3 Evaluation

In the section, we evaluate the impact of our solutions presented in Section 5.4.2 to the challenges introduced in Section 5.4.1. This evaluation is done using the firmware  $F_{\text{default}}$  and using the full experimental setup, including the USRP NF from Section 5.3.

*The connection event and the hop interval allows us to synchronize in time.*

**CONNECTION EVENT AND HOP INTERVAL** First, we use the **connection event (CE)** to precisely synchronize our acquisition trigger with the state of the **BLE** connection. Second, we empirically determined that increasing or decreasing the **CE** number at which we send the encryption **PDU** allowed us to modify the leakage position in time with a granularity of 80 ms. Moreover, it was mandatory to set the hop interval greater than 12 to have a **TX** during the **cryptographic operation** during our evaluation.

*Interleaving procedures effectively increase the leakage duration.*

**INTERLEAVED PROCEDURES COMPARISON** Section 5.4.2 introduced the technique of “interleaved procedures” leveraging the **MD** bit. We compared a non-exhaustive list procedures, illustrated by Fig. 5.10. This shows that using this technique can increase the **radio transmission (TX)** time by a factor of 2, increasing the probability of having a radio transmission from the Peripheral during the **cryptographic operation**. In our tests, the choice of the interleaved procedure impacted the leakage duration less than 10% of the maximal leakage, therefore this choice is not crucial. With more than 200  $\mu\text{s}$  of leakage, this duration was sufficient to

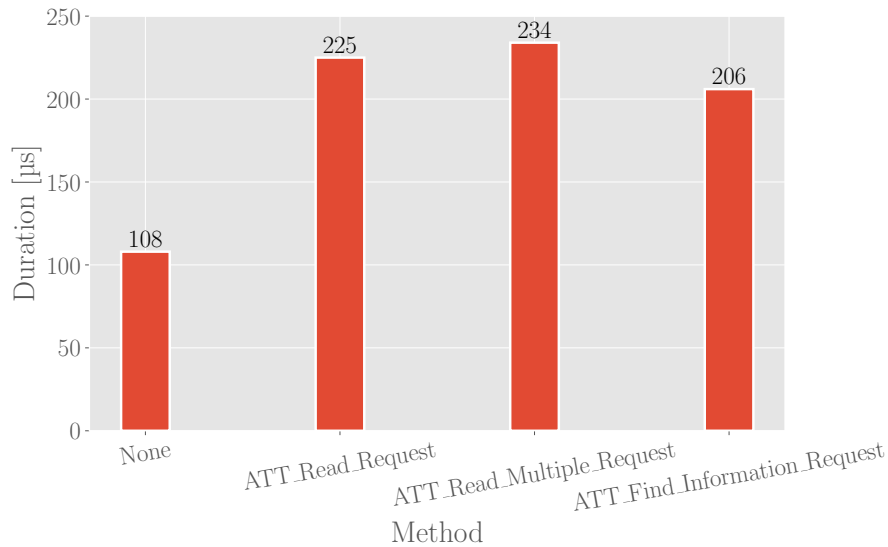


Figure 5.10: Comparison of system activity leakage duration based on procedure interleaving method.

capture a Screaming Channels leakage during most of the connection attempts.

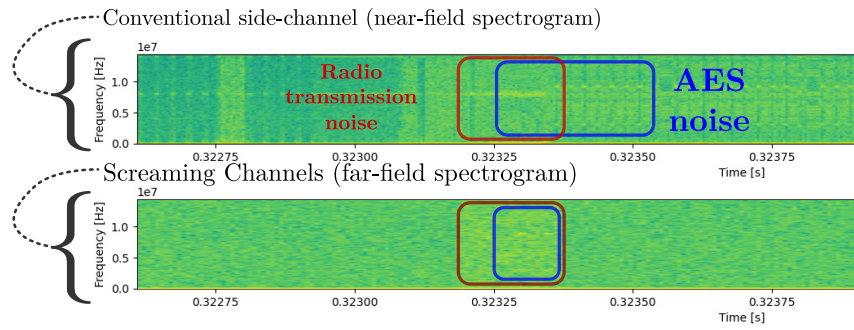
**CRYPTOGRAPHIC LEAKAGE DETECTION** The **cryptographic operation** leakage detection we performed is two-fold, with (1) An automatic detection of the AES signal inside a signal (2) A manual analysis of the system activity leakage in the **NF** and the radio transmission in the **FF**. Based on an *a priori* knowledge of the AES signal inspired from previous work [Cam+18], we used an automatic detection of the latter inside a recorded signal using frequency detection and cross-correlation matching.<sup>3</sup> Moreover, thanks to our framework, we were able to perform two parallel and synchronized recordings.

Figure 5.11a shows a comparison of two recordings. The upper one at  $2 * f_{\text{clock}}$  Hz using a **near-field (NF)** probe is recording at the 2<sup>nd</sup> harmonic of the CPU's clock, where a conventional side-channel can be recorded. The bottom one at  $f_{\text{carrier}} + 2 * f_{\text{clock}}$  Hz using a **far-field (FF)** antenna is recording at the 2<sup>nd</sup> harmonic of the CPU's clock added to the frequency of **BLE** radio carrier for Bluetooth channel number 8. On the **NF** recording, the full AES computation can always be identified (upper blue rectangle) since we have all the **electromagnetic radiation** from the inherent system activity. For both the **NF** and the **FF** recordings, we can identify when the **radio transmission** is happening (red rectangles). On the **FF** recording, only a partial AES computation can be identified (bottom blue rectangle) because the **TX** duration was not long enough – still, the exploitable part for a side-channel attack has already

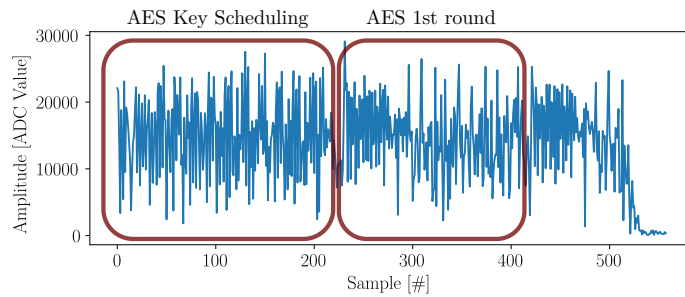
*Using the cross-correlation, we can automatically detect one AES inside a new signal.*

*Using synchronized near-field and far-field recordings, we can precisely determine at which time the cryptographic operation and the radio transmission occurred.*

<sup>3</sup> From previous AES signals captured with a custom firmware, we can use them as a reference trace to find this pattern in new signals leveraging the cross-correlation method [Rob19, p. 18].



(a) Leakage comparison between synchronized recordings of conventional **NF** (upper) and Screaming Channel **far-field** (**FF**) (bottom).



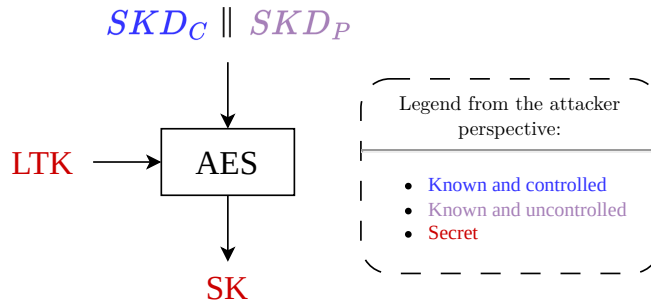
(b) AES amplitude-modulated leakage in time-domain at  $f_{\text{carrier}} + 2 * f_{\text{clock}}$  Hz (**FF**) during **BLE** communication with  $F_{\text{default}}$  firmware.

Figure 5.11: Cryptographic leakage detection.

leaked. The first goal was to identify the instant the AES computation is happening using the **NF** recording while completing a **BLE** connection procedure. The second goal was to manipulate **BLE** parameters to force this AES computation to occur during the radio transmission. To complete those goals, we leveraged both the automatic AES detection and the visual inspection – where the second goal is achieved when the red and blue boxes are overlapped similarly to Figure 5.11a.

**BLE ANALYSIS CONCLUSION** In summary, we are now able to:

- *Locate* at which time interval and which frequency bands the **cryptographic operation** will occur.
- *Force* the victim executing the **cryptographic operation** during a **radio transmission** such that the leakage will be broadcasted in the **far-field**.



**Figure 5.12:** Bluetooth Low Energy (BLE) session key derivation. The long term key (LTK) is the input key, the session key diversifier (SKD) is the input plaintext, and the session key (SK) is the output ciphertext.

## 5.5 SCREAMING CHANNEL ATTACK

We have now all the requirements to preform a Screaming Channel attack:

1. The Bluetooth Low Energy usage of the AES algorithm is vulnerable to a side-channel analysis (as seen in Section 4.4.2).
2. We have a leakage that is broadcasted through the radio receiver (as seen in Section 5.4).

In this section, we detail the Screaming Channel leakage exploitation during a BLE communication, using our framework described in Section 5.4.

### 5.5.1 Side-channel attack on AES during SK derivation

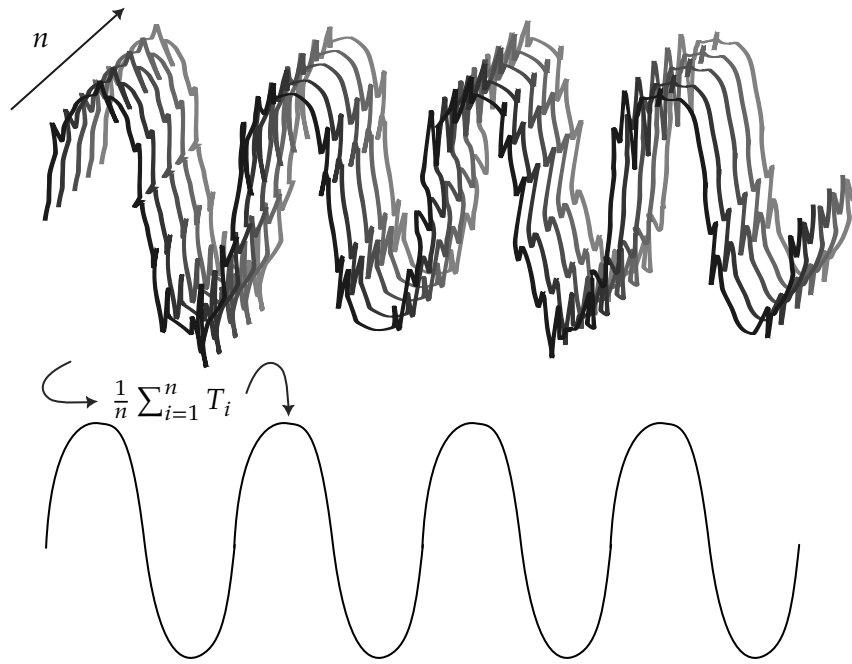
Since a Screaming Channel attack can be seen as a conventional EM side-channel coupling to a radio transmitter, we start by considering a conventional side-channel attack on BLE. Our side-channel target is the step after the first S-Box (AddRoundKey and SubBytes) inside the 1<sup>st</sup> AES execution, where the key is the LTK and the plaintext is SKD. As illustrated in Fig. 5.12 and stated in Eq. (5.1), SKD is known but partially controlled by the attacker, with SKD<sub>P</sub> chosen by the Peripheral and SKD<sub>C</sub> chosen by the Central:

$$SKD = SKD_P || SKD_C \quad (5.1)$$

As mentioned in Section 4.4, Cao et al. [Cao+23] analyzed the feasibility of a BLE side-channel attack only for the conventional case (i.e., near-field), concurrently to our work. Our work exploits the same side-channel vulnerability over the Screaming Channel (i.e., far-field), significantly increasing its impact.

*An analysis of BLE parameters and its session key derivation mechanism, we are now able to conduct a full Screaming Channel attack.*

*Since the plaintext of the AES is known by the attacker, we are able to perform a side-channel attack.*



**Figure 5.13:** Time diversity allows to improve the [signal-to-noise ratio \(SNR\)](#) of measurements by averaging a high number of traces collected under the same conditions. However, for an attack under realistic conditions, this method is no longer suitable due to the lack of control for the cryptographic input.










### 5.5.2 Profiled Correlation Attack




**ATTACK DESCRIPTION** For our side-channel attack, we used the Profiled Correlation Attack used by *Camurati et al.* [CFS20] — also known as Template Attack. In order to evaluate our side-channel attack performance, we use the [partial guessing entropy \(PGE\)](#) and the [key rank](#) metrics, described in Section 3.4.5. The training (*i.e.*, profiling) datasets are collected using a training device similar to the test device but entirely controlled by the attacker, while the attack datasets are collected on the victim device. Previous works on Screaming Channels [Cam+18; CFS20] use time diversity by averaging numerous traces (as illustrated by Fig. 5.13) and re-executing the encryption with the same parameters. This approach drastically reduces the noise for both training and attack traces, improving the profile’s efficiency. In our case, this is possible using firmware  $F_{\text{instru}}$  since we control the AES inputs. However, this approach is not possible with firmware  $F_{\text{default}}$ , where half of the input ( $SKD_P$ ) is controlled by the victim and will change for each attack trace according to the BLE protocol. The impact on the attack performance of this averaging technique is discussed in Section 5.5.5.

*We use a Template Attack without averaging traces for the realistic setup.*

**TABLE ORGANIZATION** In Tables 5.2 and 5.3, for “Profile” and “Attack” columns,  $xk * y$  means  $x * 10^3$  traces containing  $y$  AES operations each.

Table 5.2: Attack results using  $F_{instru}$ .

#	Env.	Dist. (cm)	Profile (#)	Attack (#)	PGE	Key rank	Key enum.
$A_1$	Office 	100	64k * 100	12k * 300	3	$2^{58}$	
$A_2$ 	Office 	10	64k * 300	2k * 300	1	$2^{27}$	12 s
$A_3$ 	Office 	120	16k * 300	10k * 300	1	$2^{33}$	45 m
$A_4$ 	Office 	120	16k * 300	16k * 1	0	$2^{27}$	12 s
$A_5$	Office 	120	16k * 1	10k * 300	2	$2^{54}$	
$A_6$	Office 	120	16k * 1	16k * 1	4	$2^{76}$	

When a key enumeration value is provided, this signifies that we could perform a full key recovery — illustrated by the “” icon. When reporting the [partial guessing entropy \(PGE\)](#), its median value has been chosen. We use the icons “” and “” for the omnidirectional and the directional antennas, respectively (as described in Section 5.3).

### 5.5.3 Evaluation on Firmware $F_{instru}$

Table 5.2 summarizes the conditions and the results of our attacks using firmware  $F_{instru}$  and only the USRP FF from Section 5.3. This instrumented firmware allows using the averaging technique mentioned in Section 5.5.2, denoted by  $xk * y$  with  $y$  the number of averaged AES ( $y > 1$ ).  $A_1$  attempts to attack at 1 meter inside an office environment, reducing the key rank to  $2^{58}$ . The leak is exploitable, but this attack used the omnidirectional antenna from Section 5.3 with low gain, resulting in noisy traces. To reduce the noise, we first performed the  $A_2$  attack at a smaller distance using 10 centimeters. In these conditions, we successfully recover the full key in only 12 seconds by lowering the key rank to  $2^{27}$ . To confirm the attack feasibility at a higher distance, we used the directional antenna with higher gain for attacks  $A_3$  to  $A_6$ . With a profile based on averaging training traces in  $A_3$  and  $A_4$ , we systematically perform a full key recovery. However, when using non-averaged traces for the profile in  $A_5$  and non-averaged traces at all in  $A_6$ , the performance is rapidly decreasing with a key rank down to  $2^{54}$  and  $2^{76}$ , respectively.

Averaging attack traces is not a realistic requirement for performing our attack in real-life conditions since the attacker cannot control the plaintext at every [cryptographic operation](#). However, these experiments show that noise reduction (through averaging traces or another method) is a critical requirement for a full key recovery.

*On an instrumented firmware and by leveraging averaging technique for noise reduction, we are able to conduct a full key recovery at more than one meter.*



**Table 5.3:** Attack results using  $F_{\text{default}}$ .

#	Env.	Dist. (cm)	Profile (#)	Attack (#)	PGE	Key rank	Key enum.
$A_7$ ✓	Anechoic 📶	10	16k * 1	16k * 1	1	$2^{34}$	1.5 h
$A_8$	Office 🏢	120	30k * 1	20k * 1	5	$2^{60}$	
$A_9$	Office 🏢	120	65k * 1	40k * 1	7	$2^{60}$	

**Table 5.4:** Means of [Pearson correlation coefficient \(PCC\)](#) ( $\bar{\rho}$ ) and standard deviation ( $\bar{\sigma}$ ) during profiles creation.

Attack	$\bar{\rho}$	$\bar{\sigma}$	Key rank	Full key recovery
$A_4$	0.5	0.15	$2^{27}$	✓
$A_6$	0.05	1.5	$2^{76}$	✗
$A_9$	0.275	0.8	$2^{60}$	✗

#### 5.5.4 Evaluation on Firmware $F_{\text{default}}$

Table 5.3 summarizes the conditions and the results of our attacks using firmware  $F_{\text{default}}$  and only the USRP FF from Section 5.3. This corresponds to the most challenging conditions for an attacker since no instrumentation can be used, making the averaging technique impossible.  $A_7$  is an attack inside an [anechoic](#) box with an omnidirectional antenna, isolating the setup from the environmental noise, leading to a key rank of  $2^{34}$  and a full key recovery in less than 2 hours.  $A_8$  and  $A_9$  are two attempts to reproduce  $A_7$  performance at a higher distance (120 cm), using the directional antenna. In  $A_9$ , we chained two [LNAs](#), but it did not improve the result. Figure 5.14 shows the key rank over the number of traces for  $A_7$  and  $A_9$ .  $A_7$  convergence speed is lower than  $A_9$  because we use less training traces for the profile, however,  $A_7$  converges to a lower key rank than  $A_9$  because the training traces were less noisy.

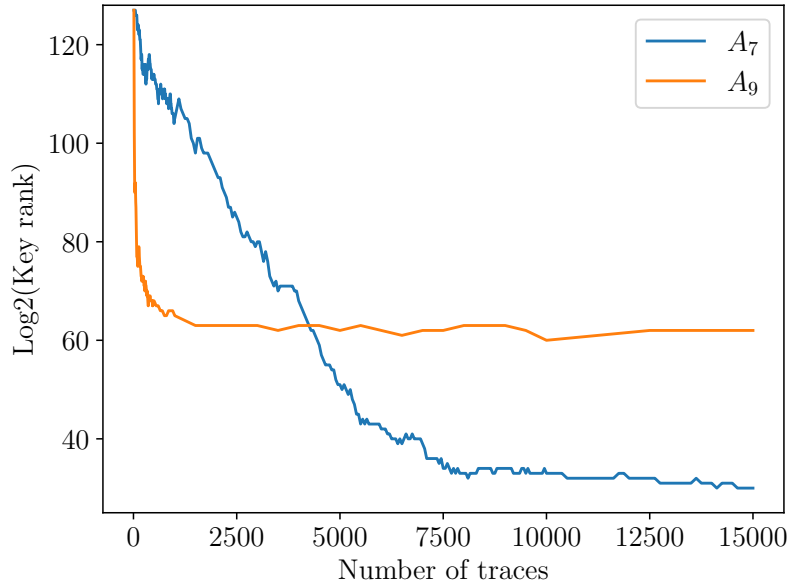
While we were able to conduct a full key recovery inside an [anechoic](#) environment, we observed a strong impact of the noise in an office environment, preventing a full key recovery.

#### 5.5.5 Impact of Noise on the Profile

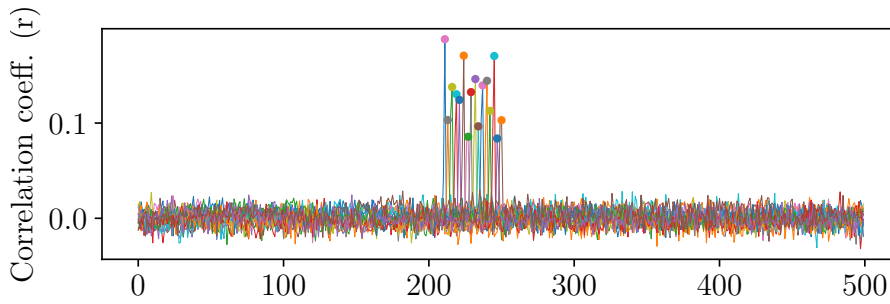
Note that each attack cannot be improved by simply collecting more traces because they reached their “convergence point”, *i.e.*, the attack performance does not increase while increasing the number of traces. This phenomenon is strongly tied to the profile quality. To build a profile, we first compute the [Pearson correlation coefficients \(PCCs\)](#) ( $\rho$ ) over the traces to test for statistical differences depending on the AES

*On a default firmware, the full key recovery is only possible inside an [anechoic](#) environment.*





**Figure 5.14:** Comparison of the key rank over the number of traces for  $A_7$  (anechoic) and  $A_9$  (office) using  $F_{default}$ .  $A_9$  is converging toward a limit due to the noise floor.



**Figure 5.15:** Correlation on amplitude for  $A_7$  during a radio transmission with GFSK at  $f_{carrier} + 2 * f_{clock}$  Hz.

inputs, as shown in Fig. 5.15. The samples with a significant coefficient correspond to the SubBytes operation of AES and are used as **points of interest (POIs)** to create the profile. To do so, we estimate the mean value and the standard deviation ( $\sigma$ ) for each possible classes, *i.e.*, the 256 possible values of  $p \oplus k$  for each subkey. We empirically observed on our profiles that the main difference between a profile leading to a full key recovery ( $A_4$ ) and a profile which does not ( $A_6$  and  $A_9$ ) is the order of the standard deviation. Table 5.4 illustrates with three representative examples the comparison between the means of both the PCCs and the standard deviation. All traces were normalized using z-score normalization [EG12; Mon+13] before computing the values. Analogous to the floor noise in radio communication, having a high-order standard deviation in our profile create a statistical floor noise, *i.e.*,

*Observing statistical parameters, we observe a noise floor in attack using profiles based on datasets without averaged repetitions.*

despite using more traces, the distinguisher cannot accurately estimate the most probable sub-key above a specific level.

# 6

## OBSERVATIONS AND CONCLUSIONS

THROUGH the previous chapters, we motivated our research project on evaluating Screaming Channels on [Bluetooth Low Energy](#). After a careful analysis of the protocol, we successfully performed a full-key recovery inside an [anechoic](#) environment, but only a partial key recovery in an office environment. In this chapter, we deliver some additional observations before discussing the results, strengths and weakness of our work.

First, [Section 6.1](#) will present additional observations and characterization of the Screaming Channel leakage. Second, [Section 6.2](#) summarize our work and discuss its relevance. Third, [Section 6.3](#) present the countermeasures that can be applied specifically against our attack. Finally, [Section 6.4](#) conclude this research project.

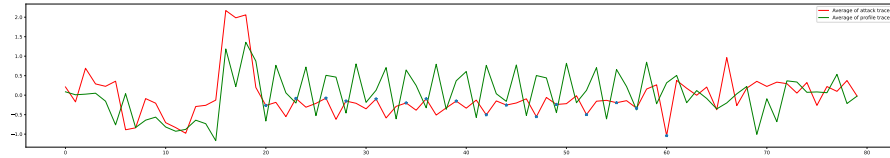
### 6.1 LEAKAGE CHARACTERIZATION

In this section, we will explore some fundamental aspects of the leakage characteristics and exploitation. In [Section 6.1.1](#), we explain which factors can impact the profile re-usage, despite that this is a commonly admitted technique in profiled side-channel attacks. Then, [Section 6.1.2](#) analyze the frequencies in which the leakage is present to deduce a presumable phenomenon leading to the leakage. Finally, [Section 6.1.3](#) expose the results of a preliminary experiment indicating which hardware component might be at the source of the leakage.

#### 6.1.1 Profile Reuse

In profiled side-channel attacks, it is common to create a profile using a controlled device in a lab and then use this profile to attack another instance of the same device in the field [[CRR02](#); [LCC08](#); [RD20](#)]. This assumes that the training device's leakage closely matches that of the target device. However, in our case, minor changes in the firmware of the target device significantly impact the leakage profiles. We faced this issue despite the use of normalization techniques, *e.g.*, z-score normalization, to improve profile portability [[EG12](#); [Mon+13](#)]. The possible root causes of those differences are, among others, static code layout, dynamic state of registers, and impulse response of the hardware reception system. In particular, we observed that:

*In profiled attacks, it is common to reuse a good profile to increase attacks performance.*



**Figure 6.1:** Failed profile reuse between two different conditions for the  $F_{\text{default}}$  firmware. The green trace is the average of the profile while the red trace is the average of the attack traces.

- The training traces recorded in the [anechoic](#) box at 10 cm ( $A_7$ ) are different from attack traces recorded at more than 1 meter ( $A_{7-8}$ ),
- The training traces recorded with the firmware  $F_{\text{instru}}$  ( $A_{8-9-10-11}$ ) are different from the attack traces recorded with firmware  $F_{\text{default}}$  ( $A_{6-7-12}$ ).

*However, using real and complex firmware, we observe that reusing profiles is more complicated than in theory.*

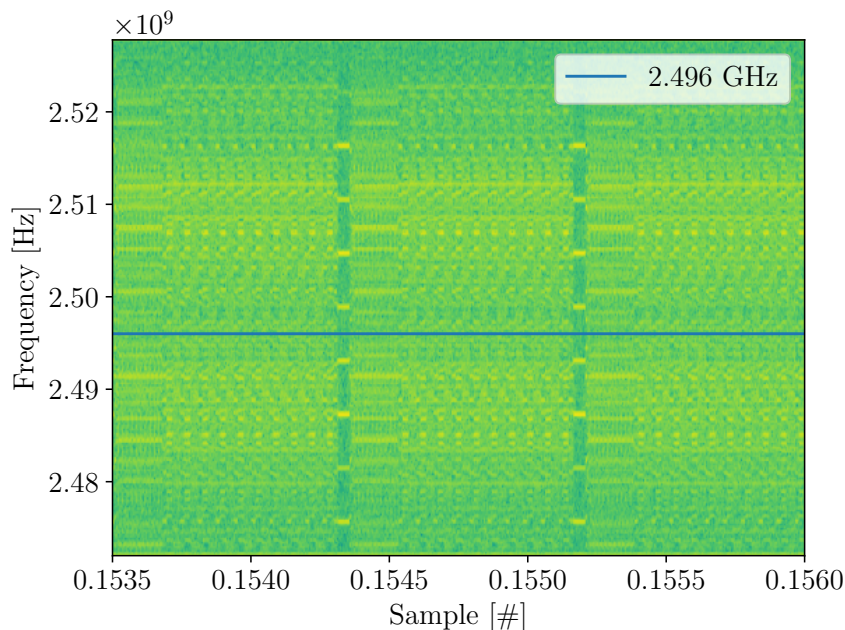
Consequently, profile reuse is complicated in attacks using either complex firmware or hardware setups. Figure 6.1 illustrates a profile reuse attempt using firmware  $F_{\text{default}}$  for both training and attack traces, but in different conditions. The profile has been created inside an [anechoic](#) environment using a small antenna. The attack traces were recorded inside an office environment using a directional antenna and an LNA. We observe that the profile mean trace (green) and the attack mean trace (red) are closely similar but do not exactly match. Not only the normalized amplitudes are different, but also the duration of the targeted step of the [cryptographic operation](#) is varying from about 2.5  $\mu\text{s}$ . Considering such differences, it is from difficult to impossible to reuse a profile between two different firmware or conditions in our case.

### 6.1.2 Leaking Frequencies

*Understanding leakage phenomena implied in the Screaming Channel attack is a complex task not sufficiently assessed.*

**PREVIOUS ANALYSIS OF THE LEAKAGE** Understanding at which frequency a leakage may be found and exploited is not a trivial task — it rely on [electromagnetic compatibility \(EMC\)](#) knowledge. *Li et al.* [[LMM05](#)] formalized and simulated how a [CMOS](#) transistor can act as an amplitude modulator, resulting in [electromagnetic radiations \(EMRs\)](#) at the harmonics of a clock signal modulated in amplitude by a data signal. *Camurati et al.* [[Cam+18](#)] described the Screaming Channel leakage as a substrate coupling between the amplitude-modulated signal from the digital part and the analog part of the SoC. While only clock harmonic frequencies were used for this initial attack, *Guillaume et al.* [[Gui+24](#)] shows that many frequencies around the [radio-frequency](#) carrier can be exploited to successfully recover a key without investigating the reason for the presence of the leak at those frequencies.

**NEW OBSERVATION OF INTER-MODULATION PRODUCTS** Therefore, we analyzed the spectrum in the frequency domain using 56 MHz of band-



**Figure 6.2:** Third-order intermodulation product at 2.496 GHz between the 32 MHz CPU sub-clock and the 2.4 GHz carrier.

width with a USRP B210 [Res], monitoring the leakage generated by firmware  $F_{\text{custom}}$  (see Table 5.1) running both AES encryption and continuous radio transmissions concurrently. We identified that the leakage was present at predictable frequencies that were not the harmonics of the CPU clock itself but the harmonics of derived clocks. The nRF52832 provides a main CPU clock (“HCLK64M”) running at 64 MHz, but also derived clocks “PCLK32/16/1M” provided to various peripherals, running at respectively 32, 16, and 1 MHz [Sem21, p. 104]. Figure 6.2 shows the signal generated by the AES leak at 2.496 GHz for a radio-frequency carrier transmitting at 2.4 GHz. The two sidebands correspond to the AES modulating the PCLK32M clock signal in amplitude. This signal has been recorded at  $3 \times 32 \text{ MHz} + 2.4 \text{ GHz}$  and corresponds to a third-order intermodulation product between the PCLK32M clock of the nRF52832 and the radio transceiver carrier. The number of derived clocks inside the SoC, and the number of harmonics and the bandwidth of the leaked signal, explains the observations from *Guillaume et al.* [Gui+24] reporting successful attacks at “non-harmonics” frequencies, which was only considering the presence of the 64 MHz clock and its harmonics. In summary, we postulate that the “non-harmonics” frequencies are, in fact, due to the harmonics of multiple internal clocks.

*By examining the frequencies of the different internal clocks of the SoC, we can predict where are the modulated clock acting as a carrier in the spectrum.*

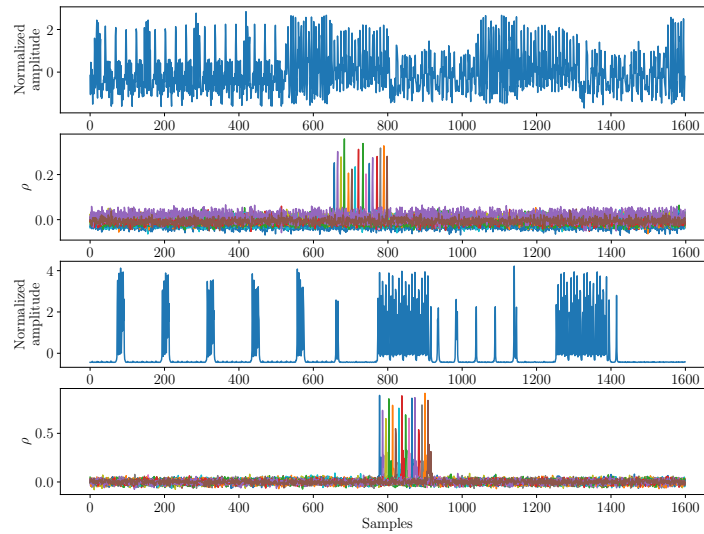


Figure 6.3: Correlations with instruction cache disabled (top) and enabled (bottom). Enabling instruction cache still allows finding correlations, but create interruptions in the leakage.

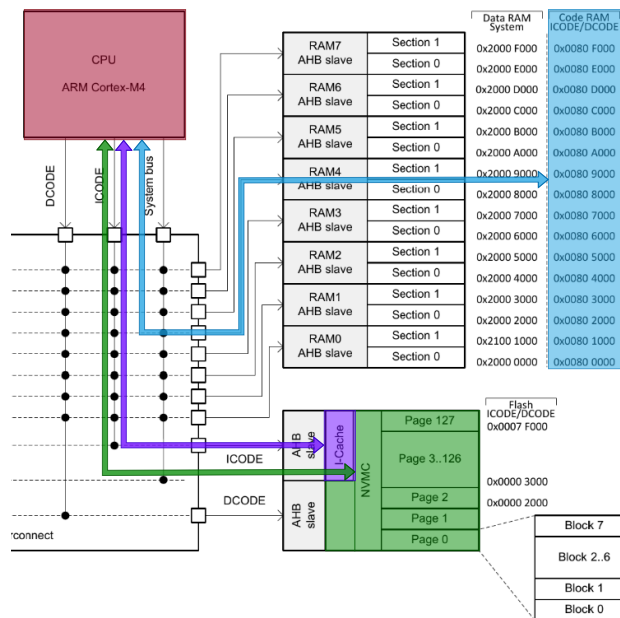


Figure 6.4: The CPU (red) can fetch instructions from either the RAM (blue), the I-Cache (purple) or the Flash (green). [Sem21, p. 24]

### 6.1.3 Hardware Component Impacting the Leakage

In Section 6.1.2, we partially depicted the phenomenon at the origin of the leakage. However, it does not incriminate a particular hardware component. Therefore, we explored the hypothesis that the memory controller (flash) may be the root cause of the leak. We observed that the nRF52832 has an **instruction cache (I-Cache)** [Sem21], allowing the CPU to fetch the firmware code to execute instructions directly from the **cache** instead of soliciting the flash memory (Figure 6.4). To evaluate the impact of this **cache**, we record two datasets of  $F_{\text{custom}}$  firmware running AES encryptions and **radio transmissions** simultaneously, with and without the **I-Cache** enabled. As depicted by Fig. 6.3, we observe that the recorded traces are intermittent. This suggests that the leakage is present when the instruction is fetched from the flash controller (*i.e.*, a **cache miss**) and absent when fetched from the **cache** (*i.e.*, a **cache hit**). We observe that the **I-Cache** is disabled on NimBLE by default, potentially for reliability reasons (for constant-time execution). Even with the **I-Cache** enabled, we were still able to correlate with the AES inputs in our experiments. These observations suggest that the code base itself, the compiler, and the linker decisions may affect the leak exploitability if the **cache** is enabled, providing an interesting direction for future work.

Overall, we observe that:

1. *Profile reuse* may be complicated due to leakage modification based on the firmware.
2. *Leaking frequencies* reveals that internal clocks are leading to inter-modulation products modulated by the leakage.
3. *Leaking mechanism* is impacted by the **cache** controller, which may not be the only involved hardware element.

Those are the results of preliminary experiments and observations, and each of these points may lead to further research.

## 6.2 DISCUSSION

In hindsight, we will discuss our methodology regarding the protocol manipulation and our evaluation. Then, we will discuss our results and its impact.

### 6.2.1 Protocol Manipulation

**GENERALIZATION TO OTHER PROTOCOLS** Using protocol manipulation from the attacker's side only, we demonstrated that it is possible to force the victim device (with unmodified real-world firmware) to transmit

*Limiting the cache controller activity also limit the leakage, suggesting an interesting research direction to identify the hardware component at the source of the leakage.*

*Low-level protocol manipulation may surely be generalized to other protocol, but this is still to be proven.*



while the target cryptographic activity occurs. While we leveraged BLE specific techniques, we identified a set of generic requirements for the Screaming Channel to happen:

- Increasing the TX duration by manipulating physical layer parameters largely increases the Screaming leakage probability. Such a low-level influence can be generated directly by injecting specific values at the physical layer or indirectly by interfering with the channel itself or triggering target retransmission mechanisms.
- Generating traffic, especially requests expecting a response from the victim device, may also lead to an increased radio transmission duration.
- Taking a fine-grained control of the timing of operations, like the control of connection events using hop interval in BLE, is also helpful for the attacker.

Moreover, our BLE manipulations are not exhaustive since manipulating other parameters could have an impact on the radio transmission duration — e.g., injecting a bigger maximum transmission unit (MTU) through its dedicated control PDU. We demonstrated this for BLE, however, for other protocols, a case-by-case analysis is required, and we expect this to be more challenging for some protocols.

*Micro-benchmarks for protocol manipulation may be used for a better evaluation.*

**INJECTION EVALUATION** Moreover, our evaluation of the protocol manipulation impact can be improved. Using micro-benchmarks to automatically characterize the impact of each modified parameter through low-level traffic injection based on numerous repetitions seems to be the most promising way. By monitoring an entire BLE connection at each run, automatically extracting both the leakage and the BLE packet signals, such technique would allow to graphically show the statistical distribution of the impact on the communication of each parameter. Leveraging our technique of automatic detection through synchronized SDRs, automated brute-force of the protocol parameters or even fuzzing — using the output of the micro-benchmark for mutating the input parameter values — may lead to an interesting direction for automated Screaming Channel attacks.

### 6.2.2 Attack Deployment and Impact

**DEPLOYMENT COMPLEXITY** Performing Screaming Channels in realistic conditions has revealed to be challenging during our work, while the feasibility and performance can be impacted by many factors. Moreover, building an experimental setup and collecting datasets require significant engineering work and take several weeks. This motivated us to release our framework as open-source software and our collected traces as an open dataset to facilitate the reproducibility of our work

*The deployment of the attack is currently a complex task, but there is a large room for improvement.*



and encourage the community to explore related topics. We identified several research directions that could significantly improve the attack performance. First, the difficulty of reusing profiles complicates the attack. Understanding the root causes of the leakage differences or significantly improving the profiling and normalization algorithms could help to reuse profiles across devices. Second, a detailed evaluation of the impact of the instruction cache could be relevant in specific scenarios. Third, evaluating the feasibility of attacking a hardware implementation of AES with Screaming Channel remains an open challenge. Finally, our experiments show that the radio setup and environment significantly impact the attack performance, indicating a need for optimization.

**A LIMITED IMPACT?** Our attack shows that Screaming Channels may be a threat against realistic firmware and off-the-shelf [Bluetooth Low Energy](#) stacks in the future. The attack still requires performance improvements before claiming that it is a fully realistic threat. Reducing the noise, which has a critical impact on attack performance, seems to be the predominant challenge. It underlines the need for a more robust radio setup, better statistical pre-processing algorithms, or signal diversity such as frequency diversity or space diversity, which are promising directions for future work. In particular, our synchronized [SDRs](#) could have been leveraged to perform frequency diversity, which we did not try at that time because of limited knowledge in telecommunication of myself. Moreover, in [Part III](#), we will introduce the amplitude-phase fusion attack technique, which also increase [EM](#) side-channel attacks performance — including Screaming Channels. We think that those two improvements could have changed dramatically the outcome of our attacks.

*With some engineering effort, Screaming Channels may become a realistic threat.*

## 6.3 COUNTERMEASURES

In [Section 10.1.1](#) and [Section 10.1.2](#), we presented general countermeasures against compromising emanations and side-channel attacks, respectively. While they are applicable to mitigate the Screaming Channel attack, we will discuss specific countermeasures against this attack at different levels.

**PHYSICAL COUNTERMEASURES** Shielding [[Wan+21](#)] or decoupling consists in attenuating the physical leakage propagation.<sup>1</sup> A hardware designer can insert a shield between the analog and the digital part of the SoC to reduce the Screaming Channel leakage. Alternatively, if the cryptographic operation is implemented in hardware, the AES S-boxes themselves can be shielded and isolated. However, shielding is

*While a perfect shield between electronic blocks would be ideal, it is often impossible.*

<sup>1</sup> The reader may refer to *Clayton Paul* reference book “Introduction to Electromagnetic Compatibility” for shielding challenges and solutions [[Pau06](#), p. 713].

inherently complicated in a fully integrated radio system that needs to transmit data over the air.

*Protocol specification countermeasures are the most straightforward and simplest mitigations.*

**PROTOCOL COUNTERMEASURES** Protocol countermeasures against side channels include limiting the number of connections in a period of time [Cao+23] so that an attacker cannot collect enough traces in a reasonable amount of time. Limiting the number of failed encrypted session establishments in a given period or per pairing or adding a waiting time after each failed session establishment also seems effective. These countermeasures are suited for a protocol specification. To counteract Screaming Channels in particular, it is essential to ensure that the **cryptographic operation (CO)** happens at random times or during the second slot of the **connection event** when the radio is disabled. These countermeasures are suited to be ensured at the firmware level.

**COUNTERMEASURES LIMITATIONS** Each of those countermeasures needs to be carefully considered, as they may have negative side effects. The physical countermeasures are quite complex and not well studied, and they may increase the cost significantly. The protocol countermeasures seem to be the most straightforward to apply and effective against this specific attack. While we need to ensure that no corner case could lead to a denial of service for legitimate users, we recommend specifying them in the protocol specification.

## 6.4 CONCLUSION

In this work, we showed how an attacker could manipulate a set of parameters of the BLE protocol to make a victim device “scream”, *i.e.*, execute critical **cryptographic operations** during a **radio transmission** while staying compliant with the protocol specification. Leveraging these mechanisms allowed us to conduct a successful end-to-end Screaming Channel attack on a victim Peripheral device in an environment isolated from noise, leading to a full key recovery of the **long term key**, used to establish a secure session between the two devices. However, assessing the attack in a realistic environment only leads to a partial key recovery due to its increased radio noise.

*While our attack performance is not astonishing, we believe that it is an interesting contribution to the side-channel research on IoT protocols.*

Future work to improve performances through profile reuse, reception diversity or amplitude-phase fusion attacks are promising directions, as it may change the outcome of an attack from a partial key recovery to a full key recovery at several meters. Regarding the improvement potential, our results led toward a threat that should be considered for the future of IoT communications.

## Part III

### PHASESCA: EXPLOITING PHASE-MODULATED EMANATIONS IN SIDE CHANNELS

In previous decades, the limits of electromagnetic side-channel attacks have been significantly expanded. However, while there is a growing literature on increasing attack distance or performance, the discovery of new phenomena about compromising electromagnetic emanations remains limited.

In this part, we identify a novel form of modulation produced by unintentional electromagnetic emanations: phase-modulated emanations. This observation allows us to extract a side-channel leakage that can be exploited to reveal secret cryptographic material. We introduce a technique allowing us to exploit this side-channel in order to perform a full AES key recovery, using cheap and common hardware equipment such as an SDR. Moreover, we demonstrate that the exploitation of this new phase leakage can be combined with traditional amplitude leakage to significantly increase attack performance. While investigating the underlying phenomenon causing this unintentional modulation, we identified several prior works that have approached similar exploitation — without being aware of each other. Creating a bridge between older and recent work, we unveil the relationship between digital jitter and signal phase shift in the context of side-channel attacks and fill the gap between prior works from various research fields.



# 7

## MOTIVATIONS

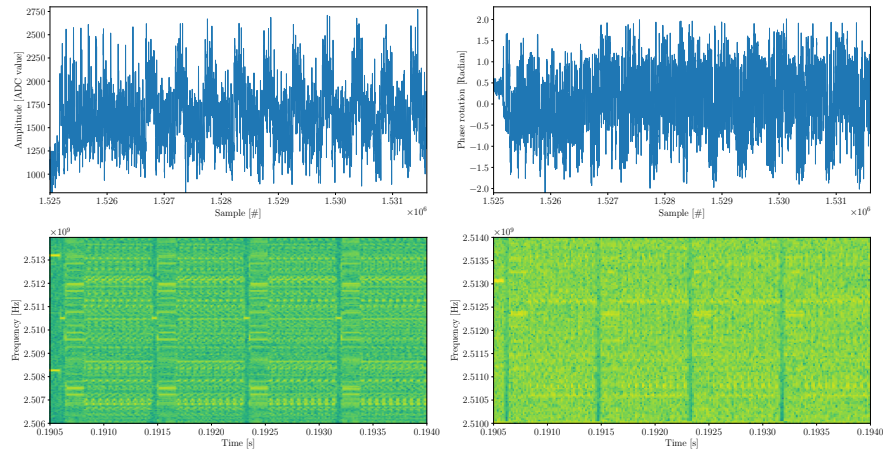
ELECTROMAGNETIC side channels have received a lot of attention recently. The biggest advantage of EM side channels is that they can be exploited using signals recorded through near-field probes placed in the vicinity of a victim device, making them non intrusive, *i.e.*, they do not need to tamper with the target device. While they are generally performed from a small distance (about a few millimeters), under specific conditions this minimal distance can be increased from half a meter [Mey12, p. 36] to several meters (*e.g.*, Screaming Channels [Cam+18]). However, in telecommunications, it is common to perform intentional modulation leveraging various parameters from the signals. Temporal parameters such as amplitude, frequency and phase are frequently used, but spatial parameters such as the polarization [Toy19] have also been explored. However, until now, the security community focus on amplitude of the radio-frequency (RF) signals for side channels exploitation. In this chapter, we demonstrate that not only the amplitude but also the phase parameter can be unintentionally modulated by side-channel leakages.

The three chapters included in this part are adapted from our publication at CHES'25 [Ayo+24b] and augmented with additional information:

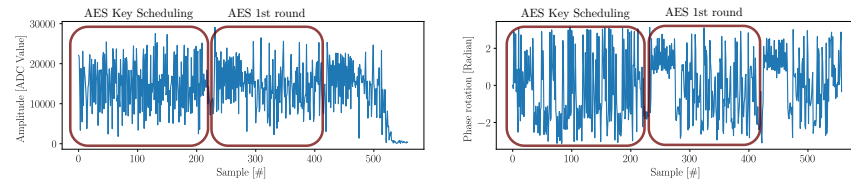
Pierre Ayoub et al. "PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels." In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) 2025.1* (Dec. 2024), pp. 392–419. DOI: [10.46586/tches.v2025.i1.392-419](https://doi.org/10.46586/tches.v2025.i1.392-419). URL: <https://tches.iacr.org/index.php/TCHES/article/view/11934>

First, Section 7.1 will relate how we discovered this new phenomenon and its relationship with Screaming Channels. Second, Section 7.2 summarize our contributions resulting of this work, from the study of the source phenomenon to its exploitation. Third, Section 7.3 will introduce the concept of signal jitter and its relation to the phase shift of a signal. Last, Section 7.4 details the differences between our work and prior researches.

*Until now, amplitude-modulated carrier is the privileged way to exploit EM side-channel attacks according to the security oriented literature.*

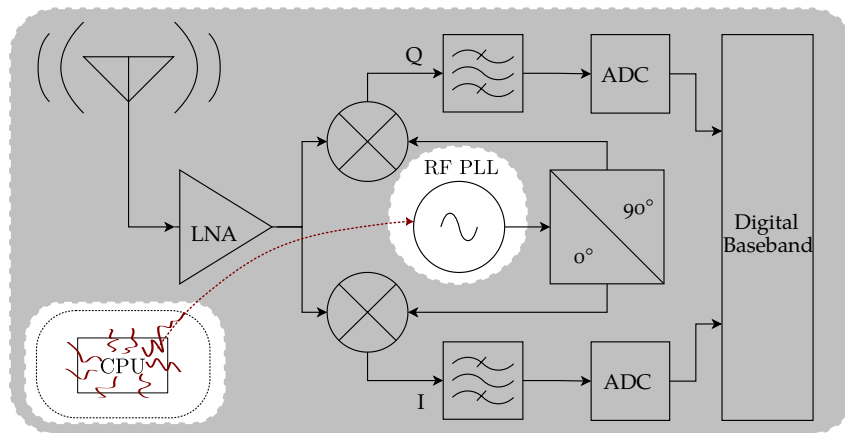


(a) Signal captured during radio broadcast from an instrumented firmware in both time-domain (upper) and frequency-domain (down) for both amplitude (left) and phase (right). We can observe the key scheduling of AES and its 10 rounds in time-domain and 4 full run of AES in frequency-domain.



(b) Signal captured during a BLE communication from NimBLE for both amplitude (left) and phase (right).

Figure 7.1: Demodulated AES leak signal from Screaming Channels frequencies in the far-field (2.5 GHz).



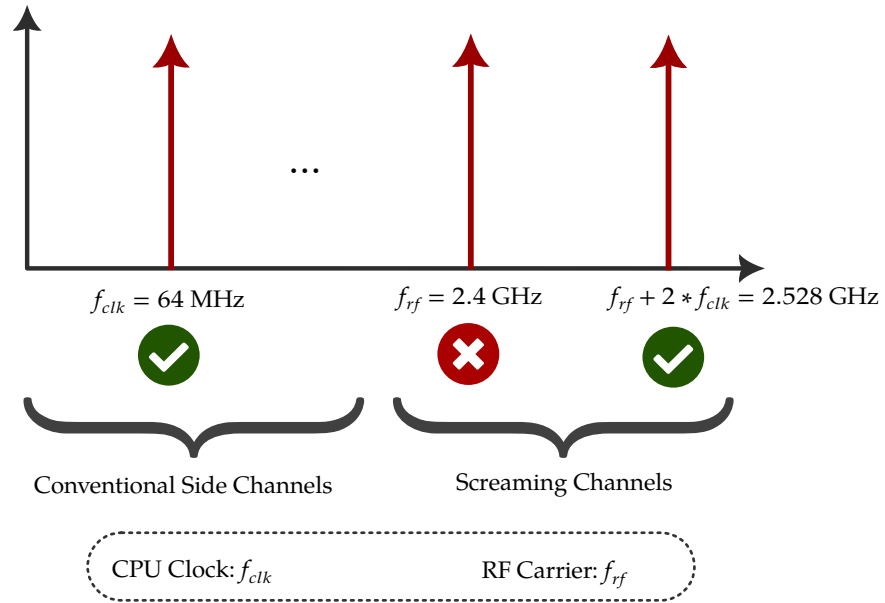
**Figure 7.2:** Hypothesis about a possible phase modulated Screaming Channel leakage. A hypothetical coupling path between the CPU and the **voltage-controlled oscillator (VCO)** of the RF **phase-locked loop (PLL)** would imply a phase-modulated output of the radio transceiver by the activity of the CPU.

## 7.1 DISCOVERY

### 7.1.1 The First Experiments

**INITIAL EXPERIMENT IN SCREAMING CHANNELS** All started with our previous research project, Screaming Channels as presented in Part II. As reported previously, the main limitation on the final attack was the amount of noise present in the traces. Therefore, we started to explore various way to improve the attack. Initially, one of our hypothesis related to Screaming Channels was that the **voltage-controlled oscillator (VCO)** of the analog **phase-locked loop (PLL)** used to generate the **radio-frequency (RF)** carrier may be impacted with the conventional leakage, as illustrated by Fig. 7.2. It implies another coupling path that would lead to a phase modulation onto the **RF** carrier generated by the digital activity leakage. By monitoring the impact of the **radio transmission** on a phase demodulator, we did not observed any impact at the frequency of the **RF** carrier (2.4 GHz). However, when tuning our **SDR** to the frequency at which the amplitude-modulated leakage was usually exploited in Screaming Channels (*e.g.*, 2.512 GHz), we observed an impact of the transmission on the phase of the signals. Figure 7.1 shows the amplitude-modulated and the phase-modulated signals during the radio transmission in the Screaming Channels context. This observation was not expected and was not coherent with the **VCO** hypothesis, because we expected to observe the opposite — *i.e.*, an impact at the frequency of the **RF** carrier but not on the amplitude-modulated leakage exploited in Screaming Channels. Therefore, we conclude that the analog **PLL** used to generate the **RF** carrier was not impacted, and that the cause of our observation may be a deeper phenomenon.

*The first phase-modulated leakage was detected when experimenting with Screaming Channels.*



**Figure 7.3:** Summary of frequency ranges containing a phase-modulated leakage by the CPU activity. The CPU clock frequency, its harmonics, and the intermodulated result in the Screaming Channel leakage contains phase-modulated leakage, but not the RF carrier.

**FURTHER EXPERIMENT IN CONVENTIONAL SIDE-CHANNEL** After this observation, we recorded the leakage in conventional side-channels setup with an EM probe in the near-field (128 MHz for our setup), without any radio transmission. We made the same observation, *i.e.*, the conventional side-channel leakage modulates the amplitude as well as the phase of our signal. The summary of which range of frequencies the phase-modulated leakage have been found during those experiments is illustrated in Fig. 7.3. At first, we suspected that this may be an artifact due to a problem in our code or in the way we performed the acquisition using the SDR. This leads us to formulate our first hypothesis about the source of the security issue: the phase modulation may be a consequence of a coupling from the digital activity to the voltage-controlled oscillator of the digital phase-locked loop (PLL). After some preliminary experiments, we understood that the effect was not specific to PLL but related to coupling with oscillators in general. Moreover, we also made the assumption that of the phase-modulated signal was not an artifact of our setup and contains a “real” information, which could be leveraged to improve existing attacks.

*We conclude that this undercovers had a great potential and that its causes and consequences needed to be systematically explored.*

### 7.1.2 Potential Impact

Our motivations to investigate this phenomenon was multifold. First, if this phase-modulated signal is a new side-channel vector, this could allow new attacks, since previous side-channel analysis did not identify



this phase-modulated leakage. If the coupling phenomenon leading to the phase modulation is stronger than the one leading to the amplitude modulation, this could improve the already existing attacks. Moreover, if the information retrieved through phase demodulation is not redundant with the information retrieved through amplitude demodulation, this could improve the already existing attacks performance, while not requiring any hardware modification for SDR based attacks. However, an important question is to know if this leakage was generalized to other *system-on-chip* or if it impacted only the nRF52832-based development kit used for the first experiments. Overall, while some questions remained at the beginning of our systematic investigation, we thought that this new leakage vector may be powerful and thought that this investigation was worthwhile.<sup>1</sup>

*Phase-modulated side channels may enable new attacks or improve existing ones.*

## 7.2 CONTRIBUTION

**UNINTENDED PHASE MODULATION** As far as we know, *electromagnetic* side-channel attacks have systematically exploited amplitude modulated signals unintentionally generated from a target device. This unintentional modulation depends on the secret data that is processed during a *cryptographic operation*, therefore leaking information that can allow an attacker to infer internal states of the cryptographic algorithm and indirectly recover the secret data. In this part, we show that another type of unintentional modulation is present in some, and probably most, *electromagnetic* side-channel signals. We highlight the presence of a leakage modulated by a *phase modulation (PM)*, a specific form of *angle modulation* caused by timing variations in a signal (so-called *jitter*). Using a radio receiver, we conduct a new side-channel attack and perform a full key recovery targeting the AES algorithm implementation (e.g., TinyAES) by taking advantage of this phase leakage. Such an attack can be performed easily with cheap radio equipment since measuring a phase can be done leveraging *software-defined radio (SDR)* costing from dozens to hundreds of dollars. In comparison, measuring the signal jitter directly requires high-end instruments, like oscilloscopes or FPGA, costing usually thousands of dollars.

*Exploiting a new side-channel vector, phase-modulated EM carrier, we are able to perform a full key recovery on AES and improve existing attacks performance.*

In this part, we follow two complementary approaches. First, we explore the feasibility of exploiting the phase information leakage on several SoCs. Second, we conduct a study of the leakage source and reproduce experimentally the phenomenon to analyze it in controlled conditions.

<sup>1</sup> Note that we did not leverage our results on phase-modulated leakage to improve our Screaming Channel attacks, since we did not know about multi-channel attacks at that time.

*Our research explore the root causes as well as advanced exploitation techniques.*

**RESEARCH QUESTIONS** To the best of our knowledge, we are the first to identify a side-channel information leakage impacting the phase of **electromagnetic radiation (EMR)**. Our work answers the following research questions:

- **RQ1:** How to detect and exploit a data-dependent leakage in the phase of a signal?
- **RQ2:** Is this phase leakage widespread in modern SoCs?
- **RQ3:** What are the physical root causes resulting in a phase leakage in **electromagnetic radiation**?

Our first approach corresponds to research questions **RQ1/2**, while the second corresponds to **RQ3**.

*Our first contributions shows that our attacks leveraging both amplitude and phase outperforms previous attacks.*

**CONTRIBUTIONS** In order to answer the previous questions, we made the following contributions:

- **C1:** We propose a methodology to compute the phase shift of a signal and generate a trace that can be processed by a standard side-channel algorithm. In our evaluation, we found statistical correlations with the cryptographic input of an AES encryption and conducted successful side-channel attacks.
- **C2:** Leveraging the state of the art on *multi-channel attacks*, we recombine independent attacks on amplitude and phase leading to an increase in performance, showing that phase leakage is not redundant with amplitude but contains additional information that can be leveraged to facilitate exploitation.
- **C3:** Using several popular SoCs (nRF51, nRF52, STM32, ATmega328), we identify leakage in the phase of signals, suggesting that the problem is widespread and not specific to a given implementation.
- **C4:** Based on prior work and our own experiments, we show that a probable cause of this leakage is a *coupling* between the processor and an oscillator circuit, producing jitter on the clock signal.
- **C5:** By explaining the relationship between a signal *jitter* and a signal *phase shift*, we fill the gap between timing and electromagnetic side-channel attacks.

*Our last contributions deep dive into the source of this leakage and how it is exploited as another form in different literature.*

Our framework is published as open-source software<sup>2</sup> and our datasets are available as open data [[Ayo25c](#); [Ayo25b](#); [Ayo25d](#); [Ayo25a](#)].

<sup>2</sup> Code: [https://github.com/pierreay/phase\\_data](https://github.com/pierreay/phase_data)

## 7.3 SIGNAL JITTER AND PHASE SHIFT

Ensuring that signal transitions always occur with precise and constant timing is nearly impossible within electronic circuits. This undesired phenomenon is called *jitter* in the electrical engineering field and applies to both analog and digital domains [HH15, p. 457]. Jitter measurement estimates the timing deviation of a given periodic signal relative to an ideal and expected one [Bal+19; Sul+90]. In digital circuits, clocking signals are typically affected by jitter effects, which are increasingly difficult to mitigate in more complex designs. This presents a significant engineering challenge as it directly affects the functional property of a design, potentially leading to data corruption or faults.

Clock jitter can be categorized based on its originating source [DS18; Hano4]. First of all, non-deterministic factors contribute significantly to the overall measured jitter. Those factors are intrinsic to electronic components, such as thermal and semi-conductor flickering noise. This is referred to as a random jitter, which follows a normal distribution. However, jitter may not be evenly distributed in some cases and could be statistically correlated to a predictable source. This non-random jitter is referred to as deterministic jitter. Those predictable jitter sources in electronics are often caused by nearby disturbing components and coupling with close signal lines. For instance, a digital data line within an integrated circuit might impact a clocking circuit in its vicinity. Thus, part of the observable jitter becomes correlated with data.

While clock jitter is naturally represented in the time domain as a period shift, it finds an equivalence in the frequency domain as phase noise [DS18; UMS02; HT03]. Considering an ideal clock signal at a perfectly constant period, it will naturally exhibit a power spectrum perfectly contained within clock frequency. Short-term period shifts added to the temporal clock signal will translate as phase noise, with frequency components spreading around the nominal clock frequency as sides-lobes on a spectrum. Overall, the use of time or frequency domain representations mainly depends on the type of measuring instrument involved [Tek05] [JG93, p. 376] – *i.e.*, a real-time oscilloscope or a spectrum analyzer. Introduced in Section 2.1.2 from Part 1, a quadrature sampling equipment with phase demodulation also constitutes a convenient way to observe jitter [Key14].

## 7.4 RELATED WORK

In this section, we will discuss the reasons researchers had to only use amplitude traces instead of phase traces until now — the reasons lying in the differences between an oscilloscope and a SDR for phase shift measurement (introduced in Section 2.1.3). We also discuss early hypotheses about unintended angle modulation by phase shift. While

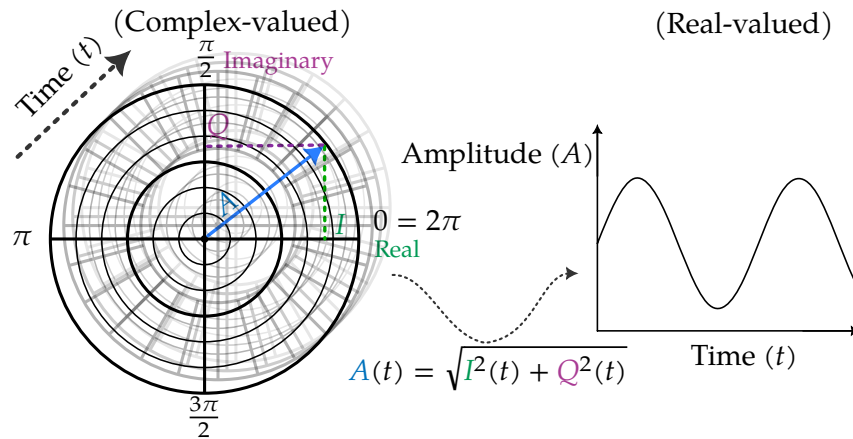


Figure 7.4: Computation of an “amplitude trace” — performing an amplitude demodulation of the complex-valued signal.

recent work has been done in exploiting signal jitter (*i.e.*, timing variations) in side-channel attacks, none of them explored its impact on phase of [electromagnetic radiation \(EMR\)](#) to our knowledge. Our motivation is to create a bridge through a side-channel perspective between jitter, a measure more common in electronics, and phase shift, a measure more common in radio electricity.

#### 7.4.1 Foundational Work: Electromagnetic Side Channels

##### 7.4.1.1 Reasons for Sticking to Amplitude Traces

To the best of our knowledge, in the current state of the art about [EM](#) side-channel attacks using radios, all of them are performed using amplitude traces instead of phase traces (*e.g.*, [[VP09](#); [Cam+18](#); [WWD20](#); [Gen+22](#)]). We explain this by two facts:

- Amplitude traces have been found to leak strongly enough to exploit them empirically.
- Working with amplitude is easier in practice since it provides an absolute measure. Computing a side-channel trace from the amplitude is straightforward, and is illustrated in Fig. 7.4. In comparison, the phase is a relative and cyclic measure that requires synchronization or post-processing, as we will see in Section 8.2.1.

Indeed, measuring the amplitude of the received signal depends on the power of the electrical signal that is fed into the ADC — hence, mainly from the antenna and the [LNA](#) gain in dB. Measuring the phase of the received signal is relative to the phase of the local oscillator that is generating the two in-quadrature signals. To have an absolute measure of the phase of the received signal with time diversity, the problem of phase coherency is equivalent to a [multiple-input multiple-output](#)

(MIMO) system, where the same signal is measured at different points in space — spatial diversity. In side channels, in order to build a dataset, we need to receive a similar signal several times at different points in time — a form of time diversity. In MIMO systems, phase coherency is typically achieved by synchronizing the clocks (local oscillators) of all radio receivers with a reference signal. However, in a side channel context, the victim and the attacker are usually not synchronized using a reference signal — it would be a strong assumption in practice. Hence, in this work, we analyze how to exploit the phase of the recorded signal from a side-channel perspective.

#### 7.4.1.2 Early Hypothesis about Angle Modulation

In *NACSIM 5000: TEMPEST Fundamentals* standard from the NSA [Ros82] declassified in 2000, there is one mention of angle-modulated carrier in EM compromising emanations. However, this document (partially redacted) do not propose any root cause hypothesis or demonstration. In 2003, Agrawal *et al.* [Agr+03] started to partially investigate the idea of angle modulation in an EM side-channel signal. This phase or frequency modulation would be caused by bad isolation between a data signal and a signal generation circuit. This work experimentally showed a frequency-modulated leakage dependent of one bit of data using a frequency-domain analysis through Fourier transform (FT). However, it has not performed an end-to-end attack against a cryptosystem and not assess the relation to phase modulation. In 2005, Li *et al.* [LMM05] emitted a similar idea, where a data line would be coupled to a VCO input voltage, resulting in angle modulation of the clock. This modulation would be visible as data-dependent timings in the time domain or frequency variations in the frequency domain. In 2011, Kocher *et al.* [Koc+11] mentioned that performing an angle demodulation before the analog-to-digital conversion may help to isolate the signal – without providing enough technical details allowing to make this claim practical. Apart from these hypotheses, to our knowledge, no paper tried to assess side-channel leakage on the phase of a signal – *i.e.*, exploiting a leakage performing an unintended angle modulation.

#### 7.4.2 Recent Work: Timing Side Channels Exploiting Jitter

Recent work has been done in timing side channels by measuring the signal jitter of clock signals. In 2021, Gravelier *et al.* [Gra+21] exploited signal jitter in delay lines to perform a remote power side-channel attack on AES. They explained that the jitter is induced by the coupling of voltage and temperature variations with power consumption. They measured the jitter using a software-based method reading registers of a delay locked loop (DLL). In 2023, Schoos *et al.* [Sch+23] published *JitSCA*, which exploits a signal jitter at the picoscale resolution of a clock

signal to perform a power side-channel attack on AES. They measured the jitter using a [time-to-digital converter \(TDC\)](#) implemented using a delay-line in an FPGA, allowing them to have a higher time resolution than a typical oscilloscope. A follow-up blog post by Riscure [[Wit23](#)] discussed *JitSCA*, and emitted the hypothesis of [PLLs](#) being one of the root cause of signal jitter. We considered a similar assumption, but we were able to demonstrate that this is a possible root cause but certainly not the only one.

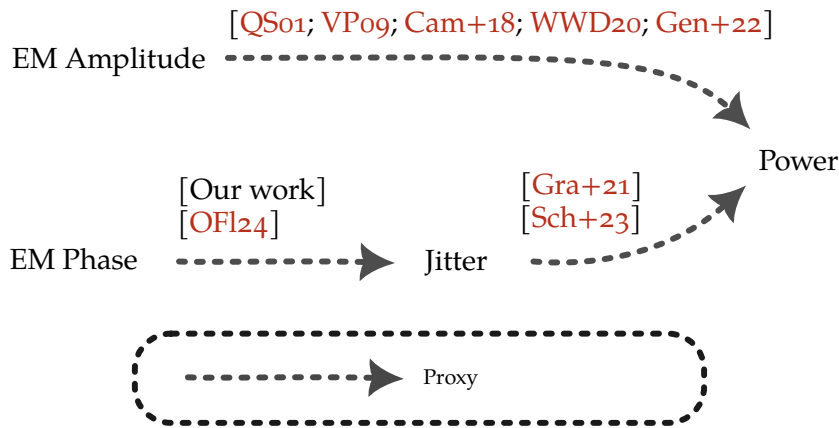
#### 7.4.3 *Parallel Work: Side-Channel Exploiting Phase Modulation*

Shortly before publishing our work, *Colin O’Flynn* [[OF124](#)] published a paper which shares similarities to our results. As our work, *O’Flynn* work tries to fill the gap between *Agrawal et al.* [[Agr+03](#)] hypothesis and *Schoos* [[Sch+23](#)] jitter exploitation. Moreover, it also demonstrates the first side channel using an unintended phase modulation on an optical link and a digital bus (JTAG). Contrary to our work regarding the phase measurement, *O’Flynn* work uses a physical connection to the targeted clock signal. In our case, we do not need any conducted measurement through physical connection nor reference clock signal since our measurement is [EM](#)-based in the [near-field \(NF\)](#). Since we need fewer requirements and use only portable COTS hardware, we believe that our measurement method is simpler to use. In this part, we propose several contributions complementary to *O’Flynn* work:

- First, we show how to exploit [electromagnetic radiation](#) measurements using an [SDR](#) through a [near-field \(NF\)](#) probe to measure the phase shift and demonstrate a method to bypass the need for a reference signal.
- Second, we evaluated the side channel on several widespread SoCs, and used a multi-channel attack highlighting that angle modulation contains complementary information to amplitude modulation from the attacker’s perspective.
- Finally, we experimentally demonstrate probable sources of data-dependent jitter and the relation between jitter and phase shift in the attacked SoCs.

#### 7.4.4 *Proxy Measurement: Equivalence to a Power or Timing Trace*

In the research of *Quisquater et al.* [[QSo1](#)] (introduced in Section [3.5](#)), measuring [EMR](#) is depicted as a proxy for power consumption measurement without galvanic conduction. The idea of a proxy measurement for power or timing traces is not specific to our work. The following



**Figure 7.5:** State of the art for proxy measurements. While it is widely known that [EM](#) amplitude is a proxy measurement for power analysis in side channels, our work and other recent work explore how [EM](#) phase and signal jitter relates to power analysis.

examples in side-channel literature use similar techniques relying on proxy measurement:

- *Spruyt et al.* [\[SMC20\]](#) showed how [fault injection \(FI\)](#) attacks can be used as a proxy measurement for power traces. Since the result of the [FI](#) is data-dependent, computing the probability of the [FI](#) success for each point in time (relative to a trigger) gives a “probability trace” which can be used in classical power side-channel attacks.
- *Nassi et al.* [\[Nas+23\]](#) showed how the intensity of the LED of a victim device recorded using a camera can be used as a proxy measurement for timing traces. Since the LED power will vary depending on the power consumption, which is flow-dependent of the executed instructions, a trace built from the LED intensity measurement can be used in classical timing side-channel attack.

While previous work demonstrates that jitter measurement is an indirect measure of power consumption in side channels, our work demonstrates that phase shift measurement is an indirect measure of jitter, as illustrated in [Fig. 7.5](#).





# 8

## PHASE MODULATED SIDE CHANNELS

PHASE-MODULATED compromising emanations has never been assessed before in [electromagnetic](#) side-channel attacks. However, this research direction was promising regarding its potential impact. In this chapter, we present our study of the root causes and the exploitation of this phenomenon.

First, [Section 8.1](#) introduce the threat model of our exploitation — although classical in [EM](#) side channels. Second, [Section 8.2](#) presents our methodology to exploits this leakage as well as studying the root causes. Third, [Section 8.3](#) exposes our different experimental setup used during the evaluation. Finally, [Section 8.4](#) shows the results of our evaluation following our methodology.

### 8.1 THREAT MODEL

We consider an attacker who aims to obtain a secret processed by a target device (*.e.g.*, a microcontroller) during a sensitive operation. In this paper, we aim to obtain the secret key processed by a cryptographic algorithm, however, the attack is generic and could impact other information types. The attacker will conduct a known-plaintext side-channel attack, recovering the key by correlating a set of measurements to a pre-computed model. The measurements are recorded using radio equipment when the cryptographic operation is occurring, in the target device vicinity (from millimeters to centimeters) using an [EM](#) probe. All our assumptions are standards for non-intrusive and passive side-channel attacks in the literature.

### 8.2 METHODOLOGY

[Section 8.2.1](#) introduces our method to compute a trace usable in a side-channel attack from the phase of a signal to address [RQ1](#). [Section 8.2.2](#) introduces the method we used to recombine information for [RQ1](#). [Section 8.2.3](#) explains how we choose several SoCs to attack and the implications for [RQ2](#). Finally, [Section 8.2.4](#) introduces our reproduction methodology for [RQ3](#).

### 8.2.1 Side-Channel Trace: From Complex-Valued Signal to Real-Valued Phase Trace

As seen in Section 7.4.1, side-channel algorithms need a deterministic measure represented as a 1D real-valued vector to work with — called a trace. While the amplitude computation is absolute and can be used as-is, the phase measure is relative, as presented in Section 7.4.1. Because the analytic signal recorded using a SDR is complex-valued, it is represented as a 2D real-valued vector. Thus, we cannot use the I/Q samples as-is for a side-channel attack. Because the phase computation is relative, it breaks the deterministic requirement for a measure, thus we cannot use the phase trace as-is for a side channel attack either.

In this section, we explain how we post-process the measured signal to generate a phase trace that fulfills the two requirements presented in Section 3.4. Figure 8.1 graphically illustrates the 3 most important steps that we present in this section to compute our trace. Such a phase trace is usable in a side-channel attack and allows us to answer half of RQ1 by testing for a data-dependent leakage.

#### 8.2.1.1 Instantaneous Phase Analysis

*The instantaneous phase is a cyclic and relative measure, not useful as-is for side-channel analysis.*

**INSTANTANEOUS PHASE** In the first step, we show how we can analyze the potential impact of system activity on the phase of our signal, exhibiting a possible data-dependent side-channel leakage. The recorded signal stored as complex numbers uses the analytic representation, as described in Section 2.1.1. We compute the instantaneous wrapped phase of our signal, the real-valued function  $\phi \in ]-\pi, \pi]$ , by taking the argument of the complex-valued function  $x(t)$  as shown in Equation 8.1 illustrated in Figure 8.1 and illustrated by Fig. 8.2:

$$\phi(t) = \arg(x(t)) = \arctan2(Q(t), I(t)) \quad (8.1)$$

The instantaneous wrapped phase represents the phase modulo  $2\pi$ , *i.e.*, constrained to its **principal values**.<sup>1</sup>

*By unwrapping the instantaneous phase, we get rid of its cyclic property and are able to compare different traces.*

**CONTINUOUS INSTANTANEOUS PHASE** The process of retrieving true continuous phase information is known as phase unwrapping [Hua+22], which leads to the  $\Phi(t)$  function representing the **continuous instantaneous phase (CIP)**, defined in Equation 8.2 illustrated in Figure 8.1:

$$\Phi(t) = \phi(t) + k(t)2\pi, \quad (8.2)$$

with  $k \in \{0, 1, 2, \dots\}$  an integer multiple of  $2\pi$  increased each time a  $2\pi$  discontinuity is encountered in  $\phi(t)$ . Therefore, the  $\Phi(t)$  is a cumulative function, not constrained to the  $2\pi$  **principal values**. Considering

<sup>1</sup> For instance, principal values of the complex argument measured in radians can be defined as values in the range of  $[0, 2\pi[$  or  $]-\pi, \pi]$ .

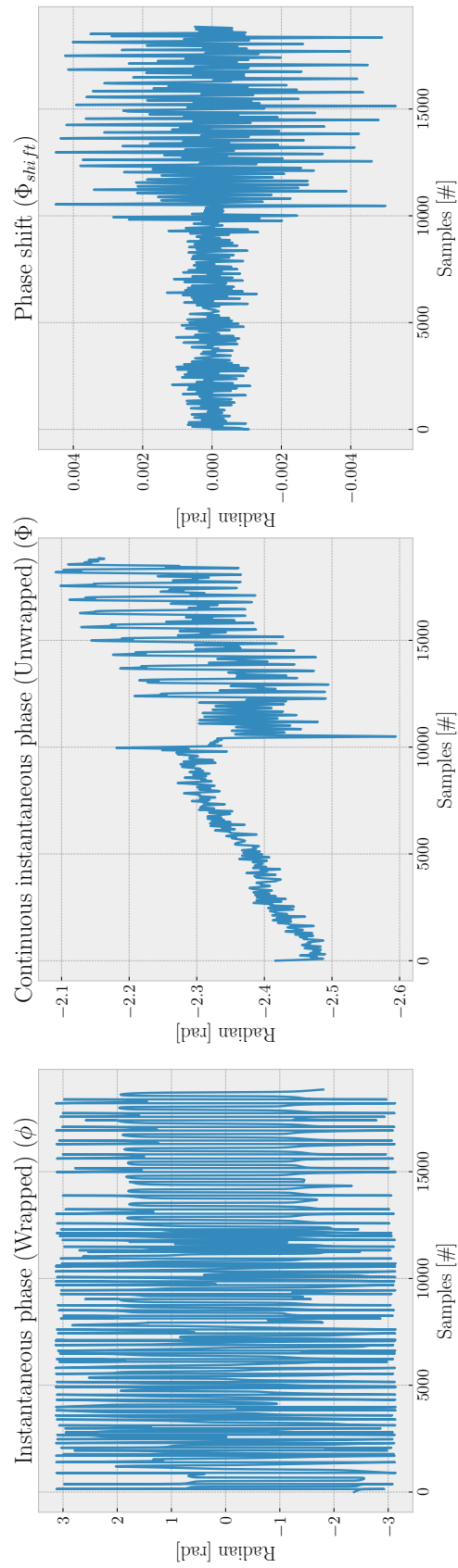
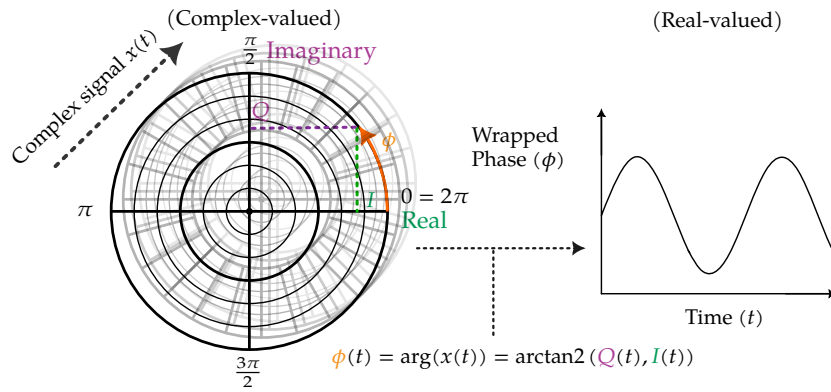


Figure 8.1: Illustration of the 3 principal steps of a phase shift trace computation (detailed in Section 8.2.1).



**Figure 8.2:** The instantaneous phase is computed by taking the argument of each complex samples. The result of this process corresponds to the first plot of Fig. 8.1.

the requirement of building a dataset to distinguish any side-channel leakage, we want to compare the CIP of multiple signals to detect a leakage. Hence, an optional step is to set our signal relative to zero to have a common starting point, as defined in Equation 8.3:

$$\Phi_0(t) = \Phi(t) - \Phi(0) \tag{8.3}$$

Using the relative to zero CIP function  $\Phi_0(t)$ , we can test if the device under test is performing an unintended phase modulation through its EMR. Indeed, if two signals have a different phase shift at some point, they will diverge. If the difference in phase shift is due to the impact of two different system activities, side-channel information may be extracted from those signals. This step is optional because only useful to compare the CIP between different signals, but is not required for the next step.

### 8.2.1.2 Phase Shift Analysis

The previous function  $\Phi$  is cumulative, which is not suitable for computing correlations and performing side-channel attacks. We must transform this function into one that will lead to a trace exploitable by side-channel algorithms.

*By deriving the unwrapped phase, we get rid of its cumulative property and are able to use it in side-channel analysis.*

**PHASE SHIFT** Instead of analyzing the CIP, we compute its first derivative, *i.e.*, the magnitude of changes in phase – the phase shift between two samples. Numerical differentiation of  $\Phi$  over time can be defined to  $\Phi_{shift}(t) \in ] - \pi, \pi ]$ , as Equation 8.4 illustrated in Figure 8.1:

$$\Phi_{shift}(t) = \frac{d\Phi}{dt}(t) = \begin{cases} 0, & \text{if } t = 0 \\ \Phi(t) - \Phi(t - 1), & \text{otherwise} \end{cases} \tag{8.4}$$

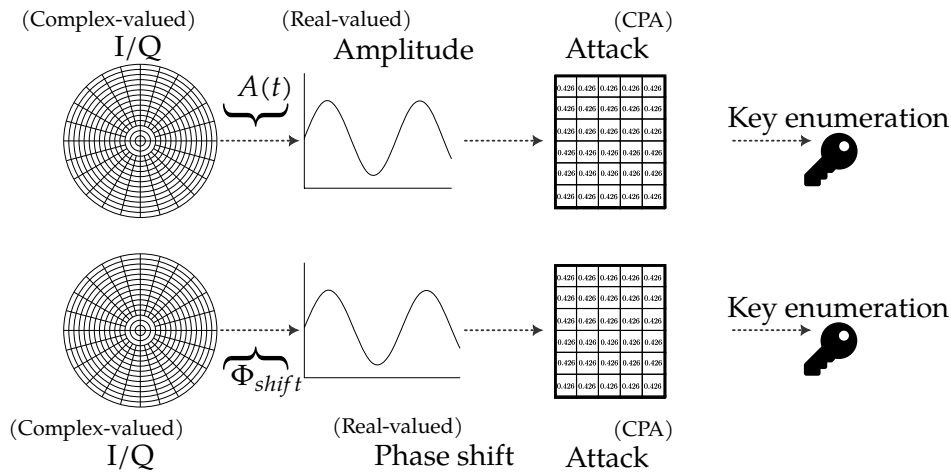


Figure 8.3: Amplitude (top) and phase (bottom) correlation-based side-channel attacks, independently of each others.

$\Phi_{shift}$  is now a function constrained to the  $2\pi$  principal values, but without discontinuities and which value represents the variation to the previous sample. It hence resolves the problem of phase synchronization exposed in Section 7.4.1.

After this transformation, the signal is now converted into a usable trace for a side-channel algorithm and can be used similarly to amplitude signals commonly exploited in state-of-the-art side-channel papers. This concludes our first part of C1.

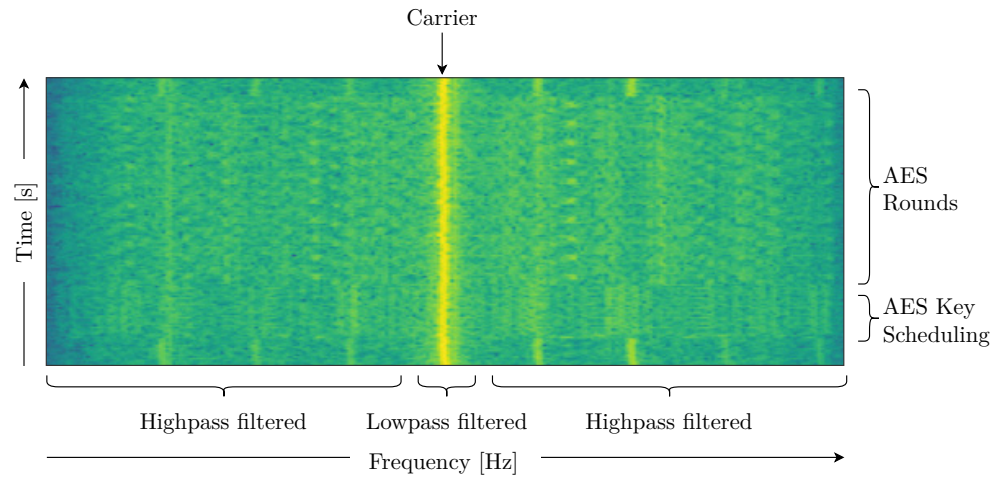
### 8.2.2 Side-Channel Attack: Using our Phase Shift Trace

Leveraging our method in Section 8.2.1, we now have a phase shift trace that meets the requirements presented in Section 3.4. First, we explain how we performed our side-channel attack using only the phase shift, corresponding to the second contribution of C1. Next, we explain how we recombined amplitude and phase information into a single attack to increase the performance, corresponding to the contribution C2.

#### 8.2.2.1 Mono-Channel Attack: Profiled Correlation Attack on Phase Shift

In this section, we will describe our attacks against amplitude and phase independently of each others, as illustrated by Fig. 8.3. Our attacks target the step after the AES S-Box (AddRoundKey and SubBytes) inside the first round.

**UNFILTERED ATTACK** For both *non-profiled* and *profiled* attacks, we first attacked without any filtering stage on our signals.



**Figure 8.4:** Waterfall illustrating filters isolating amplitude and phase shift leakage. A low-pass filter is used to isolate the phase-modulated leakage, while a high-pass filter is used to isolate the amplitude-modulated leakage.

**UNFILTERED NON-PROFILED ATTACK** For our *non-profiled* side-channel attack, we used a standard correlation attack used by Camurati *et al.* named Correlation Radio Analysis (CRA) [Cam+18], which is a variation of Correlation Electromagnetic Analysis (CEMA) [Mey12] in the context of EM analysis [QS01], itself inspired from Correlation Power Analysis (CPA) [BCO04]. In this attack, we only collect a single dataset with random known plaintexts and a fixed unknown key. The side-channel output is the key that will maximize the correlation between this dataset and the theoretical model.

*We tried both non-profiled and profiled side-channel attacks to cover a wide range of attack scenarios.*

**UNFILTERED PROFILED ATTACK** For our *profiled* attack, we used the Profiled Correlation Attack (PCA) used by Camurati *et al.* [CFS20]. First, in this attack, we build a profile using a similar device that can be instrumented before the attack, by collecting a training dataset with random known plaintexts and keys. Theoretically, the profile is created and then used across two different instances of the devices. In our laboratory setup, we used the same device for both – which does not change the validity of our results but increases performance for profiled attacks. Eventually, time diversity is possible by averaging traces where the encryption has been executed with the same input. Second, we collect an attack dataset using the target device, with random known plaintexts but a fixed unknown key. The side-channel output is the key that will maximize the correlation between the attack dataset and the pre-computed profile.

*Using fine-tuned filters, we can precisely isolate the frequencies corresponding to the amplitude or phase modulation.*

**FILTERED ATTACK** We additionally test the hypothesis that, for a given carrier frequency, the amplitude signal and the phase signal do not impact the same frequencies in the spectrum. For the amplitude signal, we make the assumption that its impact is mainly located in the sidebands

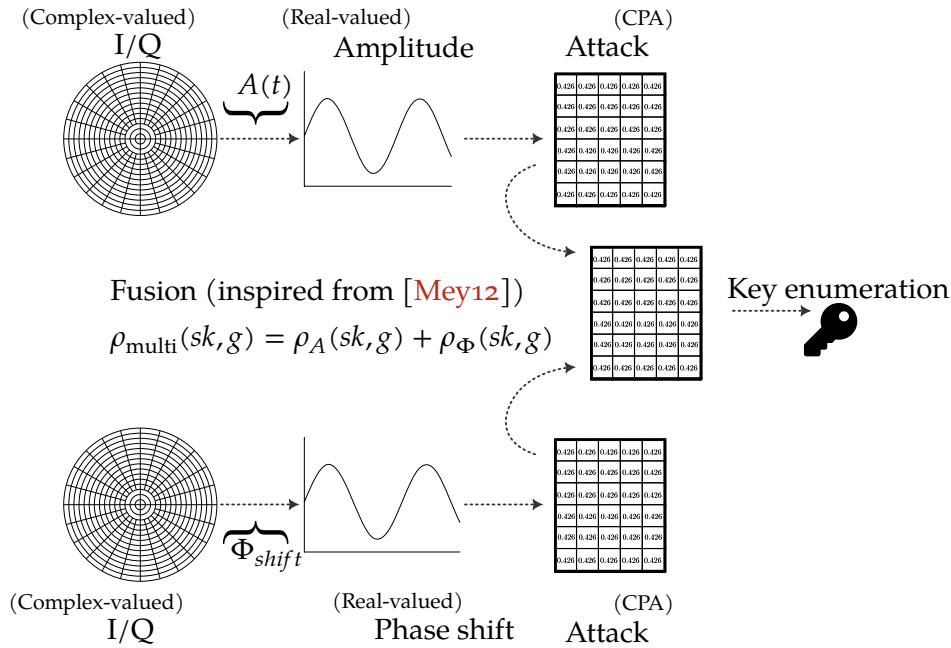


Figure 8.5: Multi-channel attack recombining amplitude and phase attack results into a single attack.

of the modulated carrier, hence, in higher frequencies around our center frequency. We make this assumption since an amplitude-modulated signal is the algebraic sum of his carrier and its two sidebands signals, in which the power is mainly distributed [Fre16, p. 99]. For the phase signal, we assume that it is close to the carrier frequency, hence, in lower frequencies around our center frequency. We make this assumption since a phase-modulated signal induce small variations in the instantaneous frequency of the carrier, in which the power is mainly contained. By using an additional pre-processing step on the complex signal, before computing the phase trace, we apply high-pass and low-pass filters, illustrated in Figure 8.4. Observing the impact of the filters on the side-channel attack allows us to identify which frequencies are mainly impacted by the side-channel information.

### 8.2.2.2 Multi-Channel Attack: Combining Amplitude and Phase in a Single Attack

Leveraging multi-channel attacks presented in Section 3.4.4, we performed recombination at the decision level, illustrated by Fig. 8.5. We chose this level because it is best suited to be implemented with our Profiled Correlation attack described above. If it improves the results, even without being the best recombination method, it is sufficient to support our hypothesis that phase and amplitude leakages may be complementary. Our technique is inspired by Meynard [Mey12], which uses the product as a combination function applied to correlation co-

*Summing our distinguishers results allows us to fusion amplitude and phase attacks.*



efficients of different time samples. We first perform two individual attacks, one using the amplitude and one using the phase, which results in  $\rho_A(sk, g)$  and  $\rho_\Phi(sk, g)$ , respectively. They correspond to the value of our distinguisher, the [Pearson correlation coefficient \(PCC\)](#), for all subkeys  $sk - 16$  in AES-128 – and possible guess  $g - 256$  values using  $p \oplus k$  as leakage variable. In our case, we use the sum as a combination function applied on PCCs of different “channels”, *i.e.*, the amplitude and the phase, defined in Equation 8.5:

$$\rho_{\text{multi}}(sk, g) = \rho_A(sk, g) + \rho_\Phi(sk, g) \quad (8.5)$$

We then use  $\rho_{\text{multi}}$  as our new distinguisher to perform the final decision. Using this recombination method, we test our hypothesis that both components can be used simultaneously to increase performance over using only one.

### 8.2.3 Generalization: Attacking multiple SoCs using Phase Shift

Leveraging our attack from Section 8.2.2, we are now able to conduct an attack on an SoC for both amplitudes, similarly to prior work and phase, using our new method. We want to know if the presence of this unintended angle modulation – leakage on the phase – was common across several SoCs. To test this assumption, we select the most popular SoCs used for microcontrollers or IoT applications, detailed in Section 8.3.1. For each selected SoCs, we record datasets using a center frequency similar to the fundamental frequency of the system clock, accurately measured using a spectrum analyzer. Some of these SoCs exhibit interesting differences regarding the separation between power domains and the generation of clock signals. Evaluating if leakage is present or not, regarding the microarchitecture of the SoC, provides an important insight into the root cause, detailed in Section 8.2.4. We also want to know if the observation of an unintentional amplitude modulation leaked from an SoC systematically results in the observation of an angle modulation leaked – or if these two phenomena are independent and can be found separately. The evaluation and answer to those questions constitute our contribution to C3.

### 8.2.4 Reproduction: Inducing Jitter and Phase Shifts in a Controlled Environment

**MOTIVATIONS** As seen in Section 7.3, jitter on digital lines can be generated due to coupling with other components in their vicinity. Previous research (introduced in Sections 7.4.2 and 7.4.3) already exploited this effect within integrated circuits to extract secret data from a processor. The influence of the temperature and supply voltage on the delays of



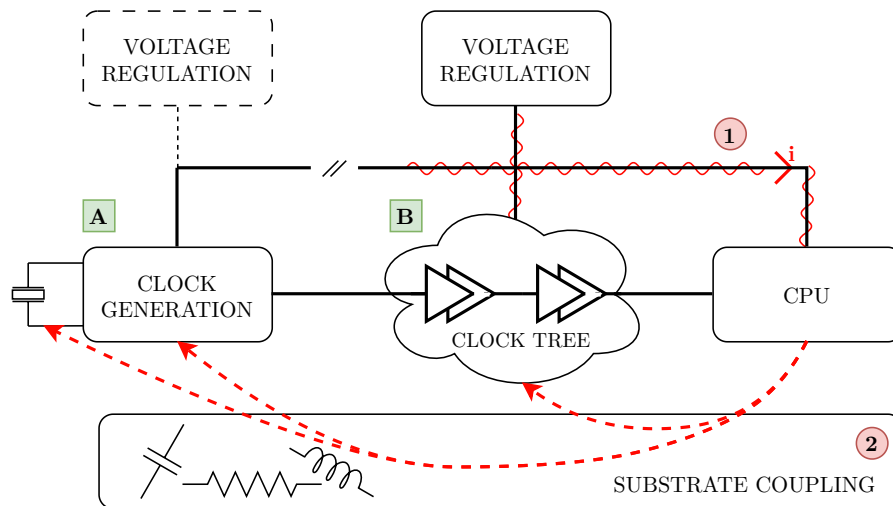


Figure 8.6: Hypothetical jitter source from processor activity.

clock transmission lines, such as clock buffers and delay lines, is often designated as a root cause hypothesis. However, this hypothesis remains they remain generalistic, which doesn't allow for a comprehensive understanding of the correlation between jitter and processor activity. Therefore, we suggest that there might be other causes that correlate processor activity with jitter. As such, this emerging type of timing side-channel can potentially be generalized to a broader range of integrated circuits, from low-end microcontrollers to highly integrated [system-on-chip](#). Aside from showing the exploitability using phase measurements on the EM field, our research focused on filling the gap between the exploitability of the phenomenon and its root causes. To do so, we empirically observed the implication of various hardware components with the observable jitter in order to answer to RQ3.

**POSSIBLE JITTER SOURCES** Figure 8.6 introduces a coarse view of a microcontroller, highlighting its clocking circuit, processor, and shared power supply. It depicts an overview of the potential sources of jitter caused by processor activity, summarizing knowledge from the literature. We denote two types of sensitive elements prone to generate jitter:

- The clock generation circuit  $\Delta$ , responsible for generating the various required clocks to the digital circuit. It generally employs an oscillator circuit for base clock generation and a [phase-locked loop \(PLL\)](#) for further clock synthesis. Both oscillators [HP00; Moh11] and PLL [Bar+02; HP01; Li18] are sensitive to voltage level variations, resulting in additional jitter on their respective outputs. The security implications of this phenomenon were previously introduced by Agrawal et al. [Agr+03].

*Clocking circuitry, including oscillators and the clock tree, are the main components prone to generate jitter.*

*Coupling between the processor and the clocking circuitry would induce data-dependant jitter.*

*By controlling the clocks and the instructions of a microcontroller, we will be able to experiment to find the root causes of the jitter.*

- The clock-tree  $\mathbb{B}$ , which is responsible for delivering clock signals to various digital components. It employs a set of buffers to balance the clock timing across the digital circuit. As such, a clock tree is similar to delay lines, as exploited by *Gravellier et al.* [Gra+21], *Schoos et al.* [Sch+23], and is affected in the same way by supply voltage fluctuations [MSY13].

The processor can affect sensitive components through a conductive (①) or parasitic (②) coupling. The conductive coupling refers to voltage variations on the power supply line due to the variable consumption caused by the processor activity. This is due to the load-regulation capability of voltage regulators [Lee99], a phenomenon usually exploited to perform power side channels. In many integrated circuit designs, the supply lines of clock generation circuits and digital logic are decoupled and provided by different power supplies. However, coupling might happen over the silicon substrate (②) due to parasitic capacitive and inductive effects.

**JITTER SOURCE STUDY** We developed a strategy to empirically confirm our previous hypothesis. The core idea is to measure the jitter of a clock signal from a chosen microcontroller while setting the latter in various configurations and states. The microcontroller's clock line can come from either a direct internal system clock or a derivative of the latter, such as a clock line from a synchronous bus (*e.g.*, SPI, I2S, JTAG). Using custom firmware, the microcontroller executes different sets of instructions, each expected to exhibit different consumption profiles. The microcontroller's internal clocking circuit should include various sensitive circuits, such as oscillators and clock synthesizers, which can be selectively enabled. Hence, we can determine if:

- 1) The observed microcontroller jitter is correlated to the processor activity;
- 2) Some internal clock generation circuits have a significant impact on the observable jitter.

For this purpose, we use an external hardware circuit that induces a controllable current flow on a microcontroller's **GPIO**. This current flow increases the overall chip consumption independently from the processor. Additionally, concurrently to jitter measurement on the clock line, the phase shift computed from the **electromagnetic** field is also measured — using the methodology introduced in Section 8.2.1. With this setup, we can determine if:

- 1) The clock jitter is correlated not only to the processor activity but also to the overall chip consumption;
- 2) The clock jitter directly translates to a phase shift observable on the **electromagnetic** field.

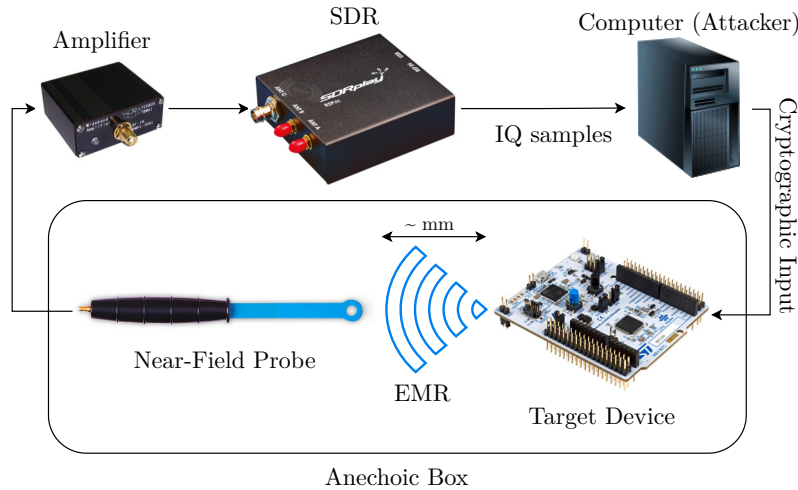


Figure 8.7: Hardware experimental setup for side-channel evaluation.

Table 8.1: Multiple target devices and SoCs evaluated in the side-channel analysis.

SoC	Ref.	Board	Clock generator	Clock freq. [MHz]
STM32L1	[STM21]	NUCLEO-L152RE	Internal RC (HSI) + PLL	32
nRF52832	[Sem21]	PCA10040	PLL	64
nRF51422	[Sem13]	PCA10028	RC	16
ATmega328	[Mic20]	Arduino Nano	RC	16
RP2040	[Pi24]	Raspberry Pi Pico	PLL	125

## 8.3 EXPERIMENTAL SETUP

In this section, we present two hardware experimental setups. The first setup presented in Section 8.3.1 is used to evaluate the side-channel attack on the phase trace for several SoCs, used for research questions RQ1/2. The second setup presented in Section 8.3.2 is used to study the source phenomenon of the unintended angle modulation, used for research questions RQ3.

### 8.3.1 Side-Channel Attack

Figure 8.7 illustrates our experimental setup used in Sections 8.2.2 and 8.2.3.

**HARDWARE** The radio receiver in use is an SDRPlay RSPdx [SDR24], allowing to record a signal down to 1 kHz, hence capturing “low frequency” clock signal of several MHz. Moreover, it allows to record 10 MHz of bandwidth at once, enabling the capture of a large band of EMR where we apply different filters in post-processing. The input of our SDR is the measured unintentional EMR of the target device using either the TekBox TBPS01 [Tek24a] or the NewAE H-Field [New24]

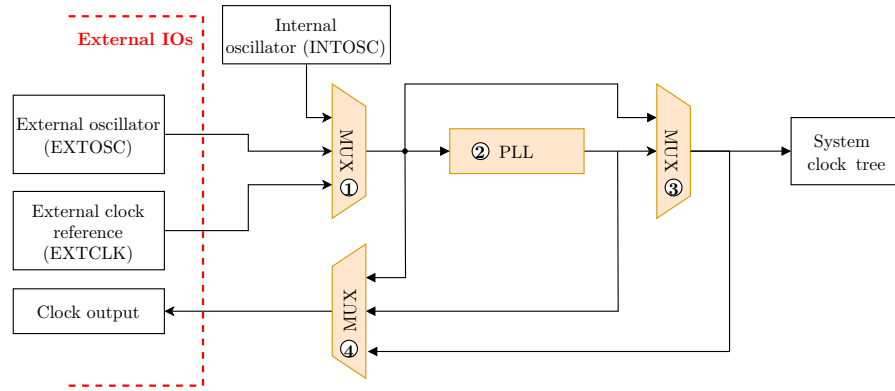


Figure 8.8: Simplified view of the STM32F103RB internal clocking circuit.

near-field (NF) probe placed at a few millimeters of the target, without galvanic contact. For some SoCs that exhibit a weak leakage, the TekBox TBWA2 [Tek24b] amplifier was used to amplify the measured EMR. The SDR is connected to a standard desktop computer through USB 3.0, sending raw I/Q during the recording. For reproducibility purposes, we isolated our measurement setup from RF environmental noise inside an anechoic box. The target device is one of the evaluated SoCs presented in Table 8.1.

**SOFTWARE** The computer is running a program controlling the target device using a serial interface. The target device is running a custom C firmware, embedding a TinyAES [Kok19] software implementation, and a hardware implementation for some SoCs (when available). In our evaluation, we focus on the software AES since its leakage is stronger than the hardware AES and the focus of the work is not to assess the difficulty of attacking hardware-based implementation.

### 8.3.2 Jitter and Phase Shift Reproduction

**TARGET MICROCONTROLLER** Our target is the STM32F103RB microcontroller [STM18] from STMicroelectronics, which is mounted on the NUCLEO-F103RB development board [STM24]. We chose this microcontroller because it offers sufficient flexibility in terms of clocking configuration to apply the methodology described in Section 8.2.4. As depicted in Figure 8.8, the STM32F1 allows switching between an internal or externally provided oscillator or clock reference (①). To provide a system clock higher than the reference one, the microcontroller uses an optionally (③) selectable PLL, whose output frequency can be tuned by the user (②). We leveraged this feature to compare the jitter effects of different sensitive clocking circuits under different configurations. More importantly, the STM32 family can route an internal clock line to an external GPIO (④). This hardware feature permits direct measurements on the internal system clock (*i.e.*, the clock fed to the digital logic),

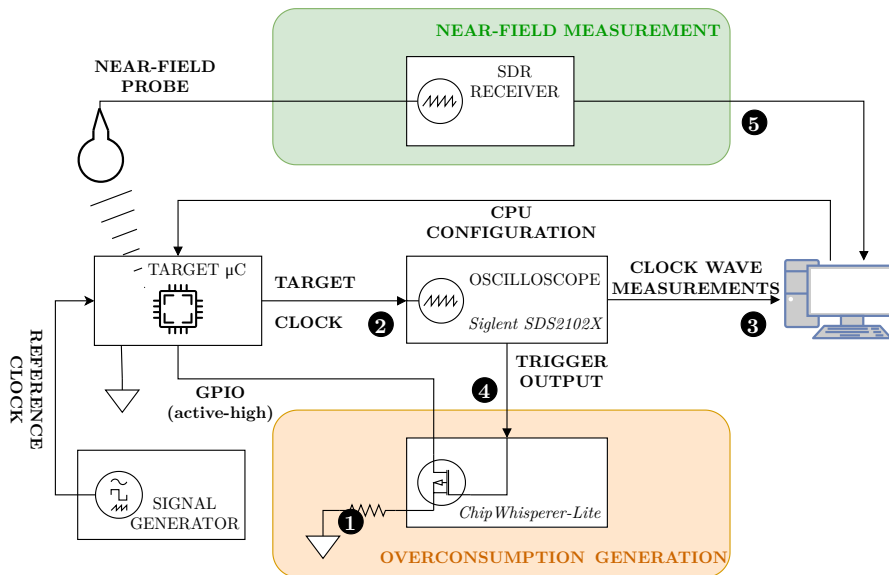


Figure 8.9: Hardware setup for jitter source study.

as well as the oscillator and PLL output clock signals. We configure the STM32F103RB with a custom firmware with TinyAES implementation (like in Section 8.3.1). Additionally, we added functions to reconfigure the internal clocking circuit on-demand at runtime. We furthermore implemented the following processor states that are expected to induce different power consumption profiles:

**SLEEP** The processor is completely powered off by clock-gating, expecting the lowest overall consumption.

**AES** The processor is enabled and continuously performs AES computation.

**STALL** An intermediate step between SLEEP and AES where the active processor executes an infinite while loop. This way, the processor only performs a branching construction without involving data memory access or arithmetic computation.

**MEASUREMENT HARDWARE** Figure 8.9 illustrates our complete testbed for jitter reproduction, implementing the methodology from Section 8.2.4. It mainly comprises a Siglent SDS2102X real-time oscilloscope [Sig20] and the STM32F103RB target microcontroller. As seen in Figure 8.8, some STM32F1 clock configurations require an external reference that we provide using the built-in signal generator function from the oscilloscope (①). Independently from other clock configurations or processor states, the microcontroller is configured to constantly output its internal CPU system clock on an external pin for further measurement. This target clock signal is connected to the input of the oscilloscope (②). The latter is configured to periodically trigger its acquisition on any clock-rising edge, acquiring as many traces as possible at a sampling

rate of 2 GSa/s. The oscilloscope is interfaced with a computer through the standardized SCPI protocol, providing continuous clock measurement traces (③). However, the communication latency does not allow retrieving enough waveforms to perform statistical computation on the jitter in a reasonable amount of time. As a workaround, we configured the oscilloscope to perform measurement and statistics of the clock period locally at the highest possible rate. This way, we collect the statistical results on the host computer only after a complete round of measurements.

To generate overconsumption independently from the processor activity, we use the glitch circuit of a ChipWhisperer-Lite [New], originally designed to conduct hardware fault attacks (④). It embeds an FPGA for fast trigger acquisition and a MOSFET to perform short circuits. The ChipWhisperer trigger input is connected to the trigger output of the oscilloscope. The MOSFET drain is connected to a [general-purpose input/output \(GPIO\)](#) from the target, which is configured to provide a continuous active-high output with an internal pull-up. On every oscilloscope acquisition trigger, a rapid short circuit of configurable length is generated on the target. This way, the oscilloscope, and the glitching circuit are synchronous, allowing us to observe the jitter effect of overconsumption on each clock trace acquisition. We reused the same setup described in Section 8.3.1 to observe the relation between the measured jitter on the clock and phase shifts in the EM field. Using the SDRPlay RSPdx, we tuned it to the configured system clock frequency, with an EM probe placed in the vicinity of the microcontroller (⑤).

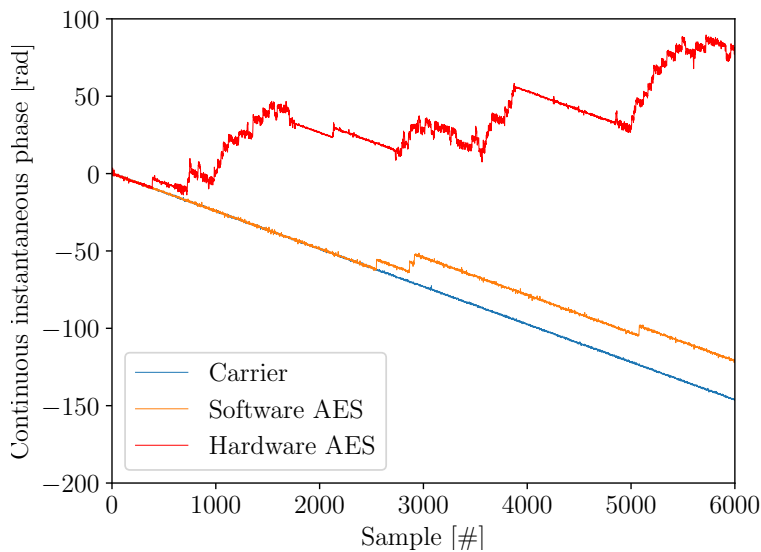
## 8.4 EVALUATION

This section details the results of our methodology presented in Section 8.2, using the experimental setup outlined in Section 8.3. In particular, Section 8.4.1 provides a preliminary answer to RQ1 and RQ3. Section 8.4.2 formally answers to RQ1/2. Section 8.4.3 answers to RQ3.

### 8.4.1 Identifying Phase-Modulated Leakage on a Target SoC

The first step before performing a side-channel attack is to identify if the target device is emanating EMR that depends on the system activity. Using our setup from Section 8.3.1 and leveraging our 1<sup>st</sup> method exposed in Section 8.2.1, we performed what we call an “Instantaneous phase analysis”.

**INSTANTANEOUS PHASE ANALYSIS** Figure 8.10 shows three relative to zero CIP functions ( $\Phi_0(t)$ ), where only the beginning of recorded signals are shown for visualization purposes. The blue signal represents the CIP of the signal at a system clock harmonic without any heavy computation

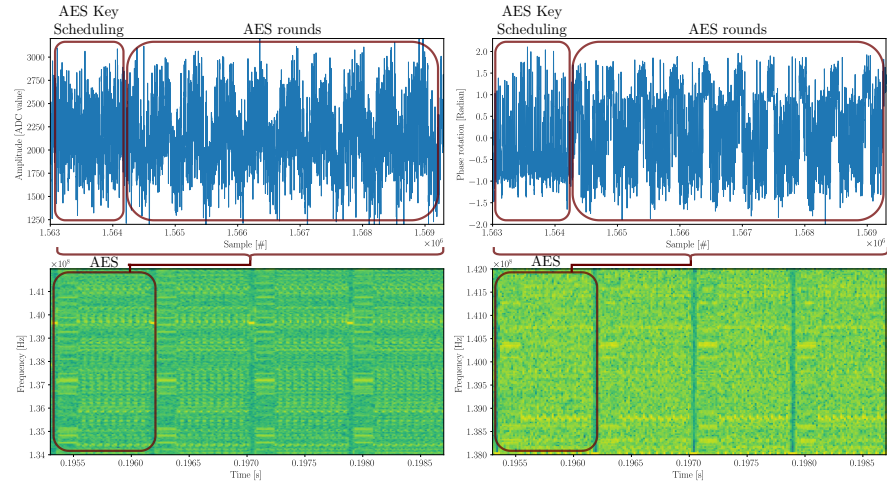


**Figure 8.10:** AES influence on the continuous instantaneous phase (CIP) of signals.

— basically, an infinite empty loop. This signal follows a constant trend across time, representing a null phase shift. The two other orange and red signals represent the CIP of the recorded signal when there is an additional system activity, respectively, software and hardware AES. We can see that those signals have variations that are not present in the first one, representing phase shifts. From this analysis, we conclude that both types of AES have an influence on the phase of the recorded signal. It raises the question of data-dependent phase modulation, implying the presence of an information leakage that may allow an attacker to compute correlations with the processed data. To answer this question, leveraging our 2<sup>nd</sup> method exposed in Section 8.2.1, we performed what we called a “Phase shift analysis”.

**PHASE SHIFT ANALYSIS** Using  $\Phi_{shift}(t)$  from Section 8.2.1 allows us to analyze the phase shift on the recorded signal generated by the system activity, more precisely, the execution of the software AES. In Figure 8.11, we show the resulting traces for the amplitude and the phase shift in both the time domain and frequency domain. This amplitude signal (left) is the one exploited by conventional EM side-channel attacks from the state of the art. This phase signal (right) is clearly generated by the AES execution, as we can identify the different steps of the algorithm. The two traces appear very similar, suggesting that the data-dependent leakage, *i.e.*, a side-channel phase modulation, is present in our recorded signal. To the best of our knowledge, it has never been exploited in the state of the art prior to this work, and this is our preliminary contribution C1.





**Figure 8.11:** AES trace in amplitude (left) and phase shift (right). Captured from an nRF52832 using a near-field probe connected to an SDR tuned at 138 MHz.

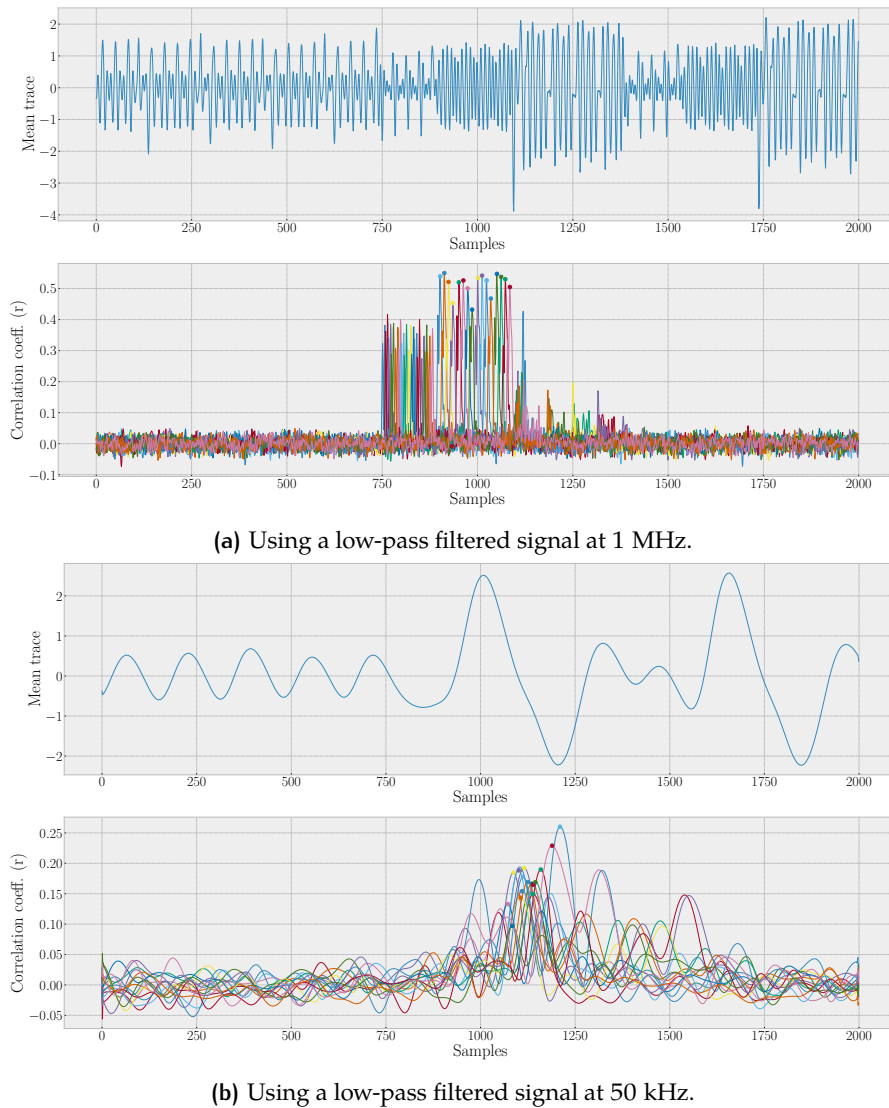
**Table 8.2:** Summary of results for evaluated SoCs in the side-channel analysis.

Legends: ✓ successful exploitation, ✗ unsuccessful exploitation

SoC	Identify AM / PM (Section 8.4.1)	Key recovery AM / PM (Section 8.4.2)
STM32L1	✓ / ✓	✓ / ✓
nRF52832	✓ / ✓	✓ / ✓
nRF51422	✓ / ✓	✓ / ✓
ATmega328	✓ / ✓	✓ / ✓
RP2040	✗ / ✗	✗ / ✗

**RESULTS ON DIFFERENT SoCs** We evaluated the presented analysis on the SoCs introduced in Section 8.3.1. The results are presented in the “Identify AM / PM” column of Table 8.2. Except for the RP2040, every tested SoCs exhibits a leakage in the phase shift trace similar to Figure 8.11. In other words, every positively tested SoCs suffers from an unintended angle modulation, which is our preliminary contribution C<sub>3</sub>. Concerning the RP2040, while the spectrum is affected by the CPU activity, we did not find any exploitable side-channel signal. This absence of exploitable signals could be linked to various factors, like a dynamic frequency scaling that could add distortion to the leakage or better isolation of the power domains reducing the leakage. The experimental results suggest that when a leakage is found on the amplitude, it is also found on the phase shift. This correlates with the results we will discuss in Section 8.4.3, indicating that both amplitude and phase shift are a proxy measurement for power consumption.





**Figure 8.12:** Correlation coefficients ( $\rho$ ) for POIs on phase shift for the nRF52.

#### 8.4.2 Side-Channel Attack using Phase Shift

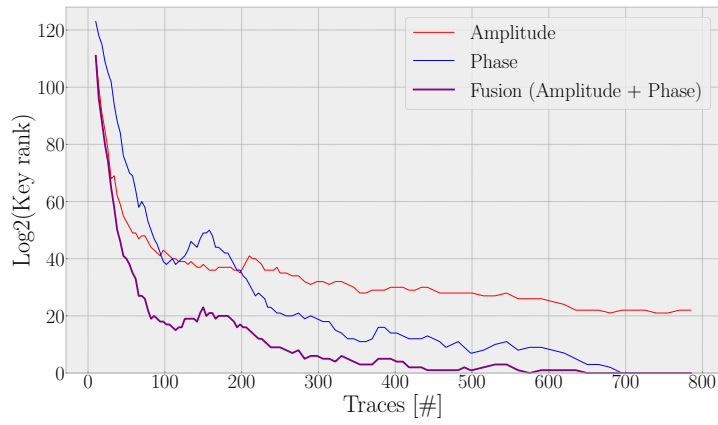
Based on the methodology presented in Section 8.2.2 and on the setup presented in Section 8.3.1, we conducted a side-channel attack using phase shift. First, we collected training datasets of 4000 traces and attack datasets of 1000 traces for the nRF52 and the STM32L1 SoCs. Training datasets were used to create profiles of the leakage with different software filters.

**SIGNAL FILTERING** Figure 8.12 is one example that shows the PCC ( $\rho$ ) between the data inputs and the measured traces in the training dataset for each sample, in order to find the **points of interest (POIs)**. It demonstrates that the leakage of the phase trace is significant ( $> 0.5$ ) on several time samples. While the signals were recorded using a bandwidth of

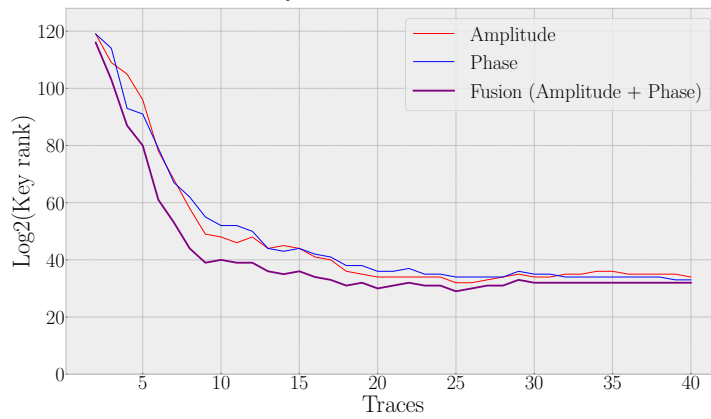
10 MHz around the carrier, filtering the signal as described in Figure 8.4 with a low-pass filter of 50 kHz leads to a profile still usable to perform a full key recovery. This experiment shows that the signal of interest in the phase shift leakage is a low-frequency signal, *i.e.*, the main frequency components are close to the carrier frequency in the frequency domain. On the contrary, we observe an opposite phenomenon for the amplitude: a high-pass filter improves the profiles and the performances, while a low-pass filter degrades them. This experiment shows that the signal of interest in the amplitude leakage is a high-frequency signal, *i.e.*, contained in the wide sidebands of the carrier in the frequency domain.

**PHASE PERFORMANCE AND FUSION WITH AMPLITUDE** Figure 8.13 and Figure 8.14 show the key rank over the number of traces for profiled and non-profiled attacks, respectively. We evaluate our attacks for the nRF52, nRF51, the ATmega328, and the STM32L1, listed under the “Key Recovery AM / PM” column of Table 8.2. The figure combines mono-channel attacks independently performed on both amplitude and phase, as well as the multi-channel attack combining amplitude and phase. First, our attacks show that using only the phase is often better than using the amplitude, *e.g.*, from an order of magnitude of 22 for nRF52 using 700 traces with profiled attack or 40 for the nRF51 using 1000 traces with non-profiled attack. Second, we observe that recombining amplitude and phase systematically perform better than attacking using them independently, *e.g.*, from an order of magnitude of 20 for the nRF52 using 100 traces with the profiled attack or using 150 traces with the non-profiled attack. This significant performance increase using a recombination method implies that, even if the radiated information originates from the same phenomenon on the emitter side, the measured information is complementary when exploiting the two components simultaneously from the receiver side.

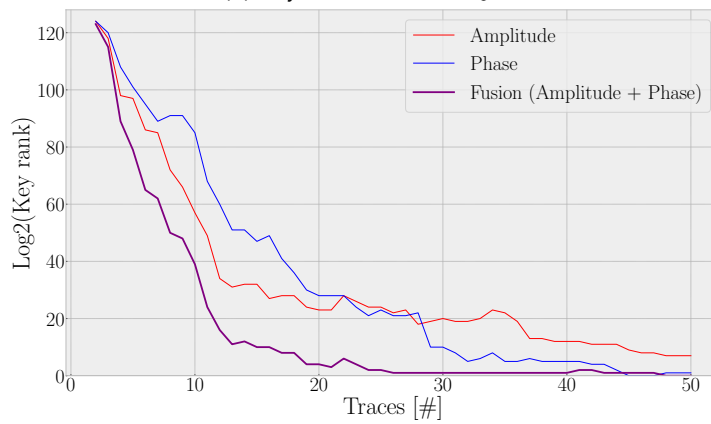
As a conclusion to C1, we can see that our method of computing a phase trace allows the exploitation of phase in a side channel attack. Concerning C2, we observe that the phase shift trace can lead to better performance than amplitude and that combining the two seems to systematically improve attack performance. It is noteworthy that we did not perform more physical measurements than state-of-the-art attacks on amplitude: we exploit the signal using an alternative approach, leading to a significant increase in performance. Finally, regarding C3, all tested SoCs except the RP2040 – which do not seem to leak, whether considering amplitude or phase – had strong evidence for the presence of the phase leakage, which we experimentally demonstrated for all of them.



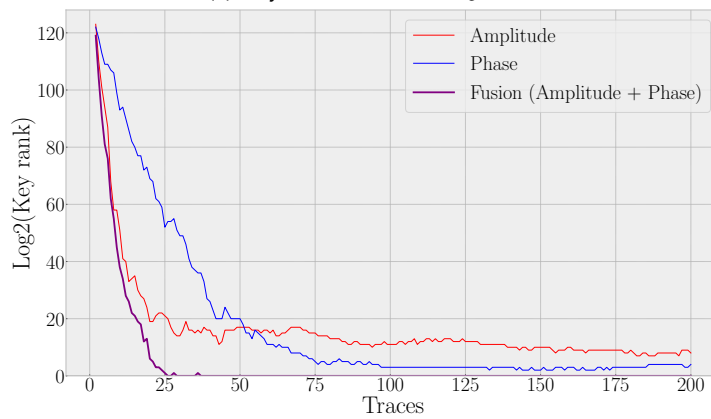
(a) Key rank for the nRF52.



(b) Key rank for the nRF51.

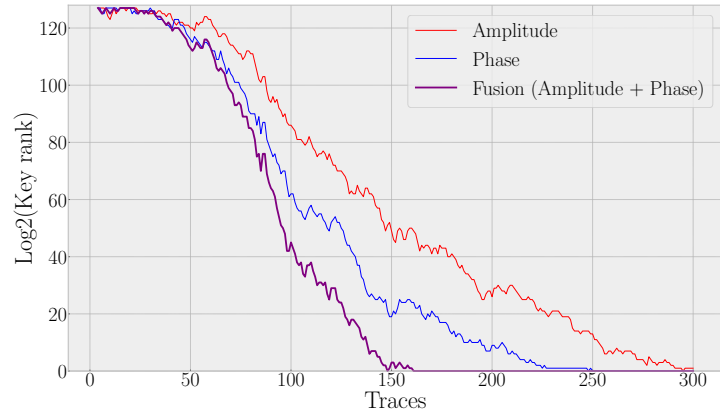


(c) Key rank for the STM32L1.

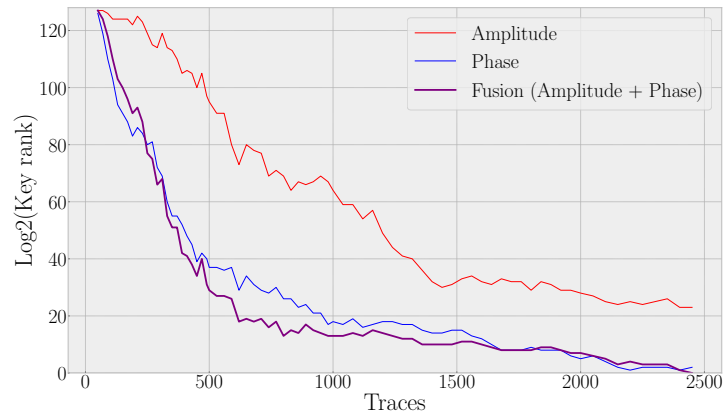


(d) Key rank for the ATmega328.

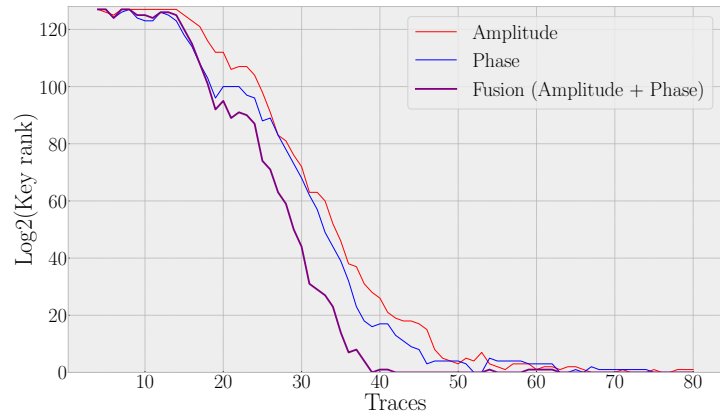
**Figure 8.13:** Performance over number of traces for profiled attacks. The smaller, the better.



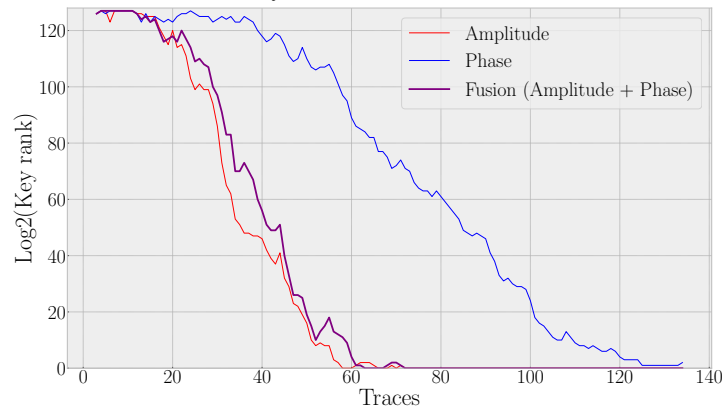
(a) Key rank for the nRF52.



(b) Key rank for the nRF51.



(c) Key rank for the STM32L1.



(d) Key rank for the ATmega328

**Figure 8.14:** Performance over number of traces for non-profiled attacks. The smaller, the better.

### 8.4.3 Jitter and Phase Shift Reproduction

Based on the methodology presented in Section 8.2.4 and setup presented in Section 8.3.2, we conducted experiments under various microcontroller conditions. For each experiment, we collected the mean, standard deviation, minimum, and maximum values of clock period measurements computed over 10,000 clock traces collection.

**JITTER VS. CONSUMPTION** We compared jitter measurements under different power consumption caused by both processor activity and overconsumption induced through a **GPIO** connected to the glitching circuit of the ChipWhisperer. Results are shown in Figure 8.15a. The mean value is represented at the center of each bar plot, where standard deviation, minimum, and maximum values are spread along. The processor's system clock is set at 64 MHz (15.625 ns period), highlighted by the central red line in the figure. Deviation from the nominal period denotes higher jitter values. Results show that the sole processor activity (blue lines) impacts the clock jitter, denoted by a substantial increase of standard deviation between the *sleep* and *stall/aes* states. We, however, observe minor differences between the two *stall* and *aes* states. Independently from the processor activity, occurring overconsumption appear to affect the jitter deviation significantly. Interestingly, we denote a mean value shifting around the nominal period in those cases. This can be explained by the fact that our observations of the clock signal occur during overconsumption events, systematically shifting the period in a specific direction. Regarding the direction of the shifts, we assume that, the rising and falling edges of overconsumption events cause transient current flows in both directions, which conversely results in positive or negative shifts of the internal supply voltage.

**JITTER VS. CLOCK CONFIGURATION** We compared the jitter measurements under different microcontroller clock configurations in the goal of determining the clock circuit that potentially contribute to the most data-dependent jitter effect. Specifically, we examined the effects of using the internal oscillator (INTOSC) versus an external clock source (EXTCLK), both with and without the **phase-locked loop (PLL)** enabled. Disabling the **PLL** limits the CPU system clock frequency to the frequency of the oscillator (8 MHz). Thus, we set the system clock to 8 MHz in all experiments to ensure the correctness of the comparison. Results are shown in Figure 8.15b. We observe that the oscillators (both internal and external) without the use of the **PLL** contribute to most of the jitter effect when the processor switches from *idle* to *aes*, with a phase deviation increase of an order of magnitude of 0.1 ns. With the use of the **phase-locked loop (PLL)**, we also denote a smaller deviation increase of around 0.04 ns from *idle* to *aes*. However, it seems that the sole **PLL** use naturally adds to most of the clock jitter without the

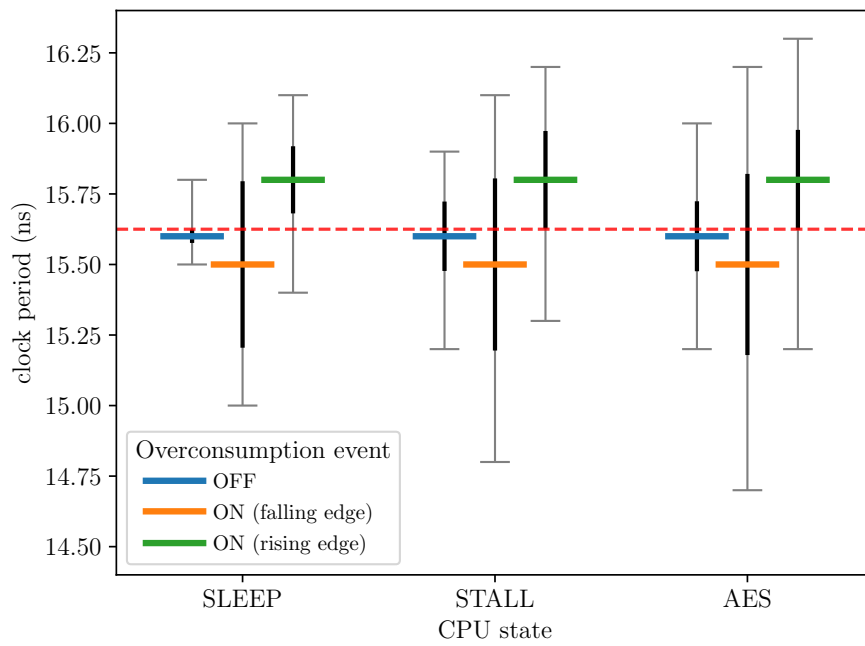
implication of any processor consumption. Overall, we observe that the various internal components contribute to both the natural jitter and the data-dependant one. However, with only the external clock source (EXTCLK), the significant jitter increase from *idle* to *aes* shows that clock circuits are not only at stake, with the hypothesis that the clock tree and paths is potentially another significant source of jitter. We keep detailed investigation of this effect for future work.

**JITTER VS. PHASE SHIFT** During near-field experiments, we configured the ChipWhisperer to produce a glitch at fixed intervals (illustrated by Figure ?? in Appendix ??). On the phase demodulated output obtained from the near-field measure, we observed periodic phase shifts of significant amplitude appearing at the same fixed interval. Equation 8.6 allows converting a timing delta ( $\Delta$ ) to a phase shift ( $\phi$ ) when knowing the frequency ( $f$ ):

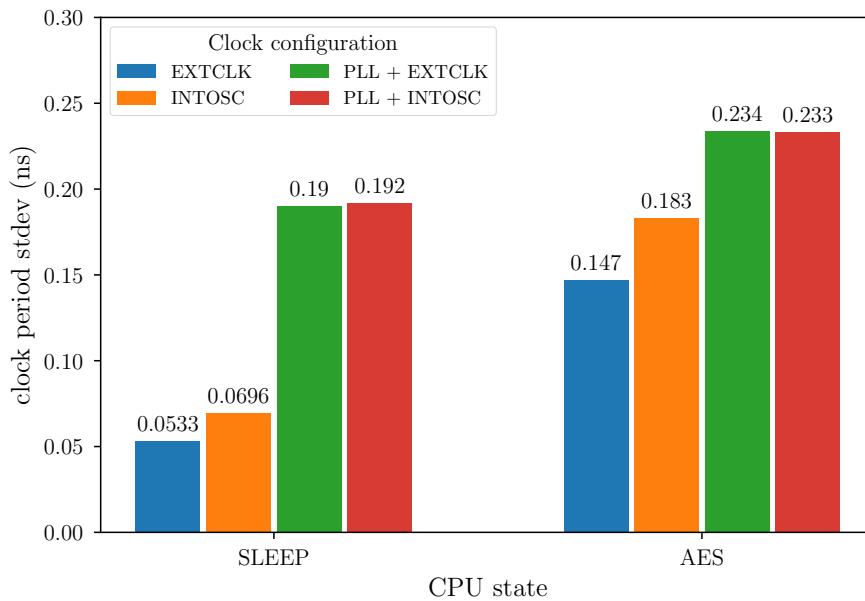
$$\phi = 2\pi f \cdot \Delta t \quad (8.6)$$

When using this formula, the measured phase shift corresponds to the jitter observed on the clock line with small incertitude. As an example, when measuring a jitter of 275 ps with the oscilloscope, we measured a phase shift of 0.125 rad, leading to an incertitude of around 0.014 rad or 11%. It highlights the direct relationship between on-chip power consumption and the phase shift observed in the near-field measurements.

During our experiments, we closely observed the clock signal behavior and jitter measurement during overconsumptions events. From that observation, we concluded that the period shift seems proportional to the short-circuit load used and, thus, the current intensity. Interestingly, we also noticed that those clock period shifts only occur during the transient effect of a short-circuit, *i.e.*, the rising and falling slopes of the GPIO voltage. Hence, the jitter effect appears to be a derivative function of the chip consumption. This is consistent with the behavior of a voltage regulator, whose voltage output sensibly reacts to a varying load over time.



(a) Clock period under various CPU states and overconsumption events.



(b) Clock period deviation for various clock configurations.

**Figure 8.15:** STM32F1 internal clock signal measurement under various CPU states, clock configurations, and overconsumption events.





# 9

## OBSERVATIONS AND CONCLUSION

THROUGH the previous chapters, we explored different aspects of the phase-modulated side channels. We show techniques allowing to exploit this side-channel vector using a standalone [SDR](#). Our work also demonstrates how to use this leakage in addition to the amplitude-modulated side channel, significantly improving the already existing attacks performances. Moreover, we designed and built an experimental setup to understand and demonstrate possible root causes of the jitter, the phenomenon producing the phase-modulated carrier. In this chapter, we will discuss the results, strengths and weaknesses of our work.

### 9.1 DISCUSSION

**NOVELTY** While we discovered this attack vector independently based on our experimental results, we also identified that phase-modulated leakage, and especially the exploitation of the root phenomenon through various techniques, was partially explored in the literature. As seen in previous sections, this leakage can take several forms, jitter or angle modulation (including frequency or phase modulation). We noted that a sub-field of the side-channel literature was already exploiting jitter, but without considering its [electromagnetic](#) effects. Moreover, as stated in [Section 7.4](#)), we recall that the experimental demonstration of a data-dependent frequency-modulated leakage by *Agrawal et al.* [[Agr+03](#), p. 8] shares some similarities with our work. However, we are the first to demonstrate the analysis of the phase-modulated leakage, and to conduct a full practical attack leveraging this unintentional [electromagnetic radiation](#). We also tried to go beyond the limits of previous papers by analyzing the root causes of the phenomenon and extending the performances of [EM](#) side channels through fusion attack.

**SIDE CHANNEL USING PHASE SHIFT** In this work, we attacked a software implementation of TinyAES, a lightweight embedded cryptographic library. While our work focuses on this software implementation, our observations lead us to consider that the phase shift leakage is indirectly due to power consumption. Additional observations also allow us to note a significant impact of a hardware implementation of AES on the phase shift. Therefore, our experiments strongly suggest that the problem is not linked to a specific algorithm or implementation. Future

work may focus on generalization of this approach by demonstrating that common targets of power side-channel attacks are also vulnerable to phase shift analysis. While we were able to observe similar behaviour on multiple SoCs using heterogeneous architectures, replicating our results on a larger set of SoCs would also be beneficial to demonstrate the widespread nature of this leakage and the benefits of the fusion strategy between amplitude and phase in side channels. Let us note that our approach to compute phase shift traces allows us to manipulate them similarly as standard amplitude traces, allowing to apply existing pre-processing techniques from prior work (*e.g.*, normalization, alignment using cross-correlation, averaging to reduce noise). The concrete impact of our contribution is significant for the field. Our approach outperforms amplitude-only attacks without additional hardware, where advantages are twofold:

- Fewer traces are required, enabling attacks where collection time is the limiting factor,
- It is more accurate for a given number of traces, improving results where accuracy is the limiting factor.

Finally, we consider our results as promising for developing new exploitation techniques targeting hardware accelerators, which are more and more common in recent microcontrollers and remain a significant challenge for practical exploitation.

**EXPERIMENTAL REPRODUCTION SETUP LIMITATIONS** We acknowledge some limitations regarding the proposed experimental setup to reproduce the phase shift. Due to the data throughput limitations of the oscilloscope, we had to rely solely on its built-in statistics computation. This limited us to the mean and standard deviation information for a given measurement set, which does not allow us to get quartiles and outliers. A more flexible setup may allow us to compute better statistics when characterizing the phenomenon.

**ADDITIONAL JITTER HYPOTHESIS** It is known from the [electromagnetic compatibility \(EMC\)](#) literature that capacitive [coupling](#) between two digital lines (a culprit and a victim) can induce propagation delay on the victim, dependent on the culprit activity [SR92; NP08]. In our case, this crosstalk-induced delay is another hypothesis for the jitter source (and hence, the phase shift). Moreover, part of the data-dependent jitter might also be due to impedance variation inside the target device, as exploited in previous work [MMT23; Kaj+23; Mos+23]. However, further research needs to be conducted to establish such a relation between our observations and this phenomenon.

## 9.2 COUNTERMEASURES

This security issue arises from fundamental phenomena studied by [EMC](#) and electronics. It is common to mitigate a specific side-channel attack by adapted countermeasures, less expensive and complex than general side-channel attacks countermeasures (presented in Section [10.1.2](#)). This is the case with our work on Screaming Channels targeting [Bluetooth Low Energy](#) protocol (presented in Part [II](#)), for which a countermeasure in the protocol specification would be the more appropriate. However, the security issues raised by unintended phase-modulation arise from an intrinsic physical problem. In this case, generic side-channel countermeasures seems to be the more appropriate. Nevertheless, in the following of this section, we will discuss jitter-specific countermeasures.

**JITTER-SPECIFIC COUNTERMEASURES** Suppose that a designer inserts a re-synchronizer circuit in the clock tree to suppress the data-dependent jitter to output a clean clock signal. First, with our attack, which measures [EMR](#), we would still be able to measure the jitter (*i.e.*, the phase shift) of the clock signal since it will radiate in the [near-field \(NF\)](#) before reaching the re-synchronizer circuit. Second, this assumes that the inserted element itself will not be coupled with the data by its power supply. For example, inserting a [phase-locked loop](#) to suppress the jitter will add another data-dependent jitter if its [voltage-controlled oscillator](#) is coupled to the data *via* the power supply. This circular problem may be solved with proper isolation between data processing and any clock generation circuitry. *Yu et al.* [[YK18](#)] evaluated countermeasures at the voltage regulator level. They reduced the correlation coefficient up to only 80% for a [DPA](#) by masking the data-dependent leakage using voltage and frequency scaling. Therefore, this seems to be a hard problem due to phenomena like, but not limited to, substrate coupling [[Par09](#)], for which we do not have proper countermeasures.

## 9.3 CONCLUSION

In this paper, we demonstrated for the first time an [EM](#) side-channel attack exploiting unintentional phase modulation instead of amplitude modulation. We were able to conduct this attack without a galvanic connection to the target by measuring the signal using an [SDR](#) and a near-field probe. Our attack allows us to successfully recover full AES keys only using the phase of the signal, highlighting the relevance of the signal phase for conducting side-channel attacks. Moreover, we applied a fusion attack technique, allowing us to combine both amplitude and phase information to significantly improve the attack performance. This approach outperforms amplitude-only attacks without additional

hardware, resulting in an improvement by an order of magnitude of 40 in the best scenario. Indeed, with our approach, fewer traces are required to conduct an attack or attacks are more accurate for a given number of traces. Moreover, by evaluating the leakage on five popular off-the-shelf SoCs, we identified that four out of five suffer from unintended phase modulation and were able to exploit this unintended modulation to conduct a successful key recovery. Finally, we investigated the root causes of this phenomenon by designing an experimental setup to reproduce the physical problem inducing the phase leakage. We highlighted the relationship between data-dependent jitter on clock signals and EM phase leakage caused by variations in the power consumption correlated to the processor activity. Based on our experiments and a comprehensive literature review, we filled the gap between the early hypothesis about this leakage introduced twenty years ago and the recent timing-based jitter side-channel attacks, paving the way for a better understanding of the phenomenon and the development of new offensive techniques.

## Part IV

### PERSPECTIVES

Compromising emanations and side-channel attacks are long-term security issues. Despite being studied for several decades, our work shows how unanticipated attack scenarios can be exploited and the identification of new leakage sources. There is evidence to claim that these security issues need to be addressed from the hardware design to the software implementation as well as protocol specifications.

In this part, we will first give an overview of the countermeasures relating to our contributions and discuss them. Finally, we will depict future promising research directions and conclude our work.



# 10

## COUNTERMEASURES

**S**ECURITY issues arising from the interactions between multiple layers must also be mitigated at multiple layers. Depending on the layers, the countermeasures can be more or less easily deployed or efficient. For example, some countermeasures may be difficult to implement in practice, depending on the complexity of the targeted software and hardware components and layers. In this chapter, we will first review the different layers at which a countermeasure may happen in Section 10.1. Then, we will discuss the nature of these security issues and the author’s view about designing secure systems in Section 10.2.

### 10.1 TAXONOMY

We can identify four levels at which a countermeasure may be applied to mitigate [electromagnetic](#) side-channel attacks and [compromising emanation](#). In this section, we will first give examples of countermeasures for each layer — and discuss their feasibility, cost, and efficiency. However, we will only discuss generic countermeasures — for specific countermeasures against our attacks depicted in Part II and Part III, countermeasures are described in the chapter directly.

#### 10.1.1 *Physical*

Leveraging the knowledge of [electromagnetic compatibility \(EMC\)](#) introduced in Section 2.2 from Part I, it is possible to consider countermeasures at the electronic level against [electromagnetic](#) leakage. This set of techniques has the goal to either, reduce [electromagnetic interference](#) and [crosstalk](#) such that the red signal does not propagate to the black line, and reduce the amount of [electromagnetic radiation](#) such that the leakage cannot be captured by the attacker. In the following, a non-exhaustive list of possible countermeasures is described [[HH15](#), p. 579]:

**FILTERING** Filters with accurate frequency-response must be employed in order to limit the [crosstalk](#) between lines [[HH15](#), p. 391]. Hardware designers can use both passive (*e.g.*, RC or LC filters) or active filters (*e.g.*, using MOSFET switches). While filtering must be systematically used in secure chips, it is unlikely to be perfect and thus cannot prevent unintentional coupling on its own.

**SHIELDING** Shielding is achieved through a metallic enclosure blocking **electromagnetic radiation** [Pau06, p. 713] [Devo8a]. In a security context, its purpose is two-fold: blocking the **compromising emanation** from radiating outside of the **SoC** generated from the internal activity, and blocking external **electromagnetic radiation** which could be used in an illumination attack. However, by definition, an embedded device transmitting information through **electromagnetic wave** cannot be entirely shielded.

**GROUNDING** Several grounding strategies exist [Pau06, p.796]. For example, in the *single-point* grounding strategy, each subsystem has its own current return path to ground. However, unintentional capacitive or inductive coupling may still happen between the wires or the **PCB** tracks, and this strategy introduces significant routing complexity. In a *multi-point* grounding strategy, each subsystem is connected to a common current return path. However, resistive coupling is likely to occur in this context. Moreover, when a difference in voltage occurs between two ground points, a so-called *ground loop* problem will occur — where the two points will act as a voltage source and will drive *common-mode currents* between two hardware blocks. In summary, grounding must be carefully assessed during chip design and compliance tests, including security tests and not only functional tests.

**DECOUPLING** In its general meaning, decoupling aims to “prevent coupling”. However, in its most used meaning, it corresponds to preventing “resistive coupling”. It is mostly achieved *via* decoupling capacitors [Pau06, p. 838], acting as a reservoir of energy. For each hardware module, a decoupling capacitor can be connected between power and ground pins. It will allow the hardware module to draw current from the decoupling capacitor instead of the power rails during the switching activity.

**PCB DESIGN** Another important aspect is the **PCB** design and the component placement. For example, keeping the highest-speed components at a distance from external connectors [Pau06, p. 807] allows to limit potential coupling. Another strategy aims to keep the highest-frequency components closer to the center of the **PCB**, to take advantage of its intrinsic filtering [Pau06, p.831]. While **PCB** component placement is mostly achieved automatically today, those rules should be incorporated into layout tools as strict security countermeasures.

The techniques discussed above have been known for a while in the **electromagnetic compatibility (EMC)** community. However, we consider that it is an important challenge for the electronic and chip designers to leverage those techniques with security compliance tests in addition to functional compliance tests. While security compliance



tests are already performed by governments or intelligence agencies for military or defense communications, such as NSA in the USA or ANSSI in France, we consider fundamental to create a bridge between hardware security researchers and private companies in order to fill this gap. Without such a coordinated effort, [compromising emanation](#) will still continue to be a threat to end users, and may even increase with the massive expansion of [Internet of Things](#) devices — since the [system-on-chips](#) are more and more integrated and complex.

### 10.1.2 Cryptography

Cryptographers are implied in side-channel attack countermeasures. In fact, a side-channel attack is enabled by a direct correlation between a leaking signal and an intermediate value, which is only dependent on the secret key. Therefore, by suppressing the direct correlation between the leaking signal or suppressing the dependency of the intermediate value on the secret key, it is possible to mitigate side channels at the information level. Without being exhaustive, we highlight two well-known countermeasures requiring changes in the cryptographic algorithm implementation itself to prevent side channels:

- *Masking* [[Sha79](#); [PR13](#)][[MOP07](#), p. 223]: The secret is divided into multiple “shares”, mixed with true random values during intermediate cryptographic computations and unmixed at the end. Therefore, an attacker cannot infer information about intermediate values because of the random mixing. This technique allows to suppress the dependency between the intermediate value and the secret key. However, it is based on the assumption that the *true random number generator (TRNG)* is not flawed — however, some physical attacks may allow to bias them [[GR20](#), p. 8].
- *Hiding* [[LH20](#)][[MOP07](#), p. 167]: The execution time and the power consumption is carefully controlled during cryptographic operations to appear random or constant, allowing to design balanced algorithms. Therefore, an attacker will not be able to gain information through side-channel measurements because the direct correlation between the leaking signal and the secret key is suppressed.

While those countermeasures are often implemented inside some secure chips, it has some disadvantages. First, cryptographic algorithms are increasingly implemented in hardware, because of performance and energy saving reasons. Then, implementing such countermeasures in hardware might add too much complexity to the hardware design for low-end or low-energy devices. Moreover, with time constrained protocols such as [Bluetooth Low Energy \(BLE\)](#), the performance penalty may not match the strict timing constraints required by the protocol.

Second, it is known today that only applying countermeasures to cryptographic algorithms cannot be the only solution. For example, a recent work from Danieli *et al.* [Dan+24] on expending Screaming Channels demonstrates that it is possible to retrieve raw data transmitted on hardware buses at distance — such as RAM buses, UART, JTAG, or SPI. Considering a Secure Element that is holding a key that must be transmitted to the central processor using a non-encrypted bus, leveraging the techniques demonstrated in this work, the key might be retrieved by an attacker at a distance by leveraging [compromising emanation](#). As such, cryptographic algorithms have to be protected, but we cannot achieve a complete security without controlling the [compromising emanation](#) *via* the techniques presented in Section 10.1.1.

### 10.1.3 Specification

**SECURITY PROPERTIES** The very first “countermeasure” that specification should apply is *non-ambiguity*. Effectively, during the last decade, it has been shown that some attacks exploited specification ambiguities leading to flawed implementations. One example is the ReVoLTE [Rup20] attack in LTE networks, which exploits keystream reuse because of an ambiguity in the specification defining nonce re-generation. Moreover, good practices often imply the use of encryption for *confidentiality*, signature for *authentication*, and hash functions for *integrity*. The lack of one of those properties may break the others, such as in the ALTER [Rup20] attack on layer 2 of LTE, which breaks the confidentiality property because of a missing integrity. However, those good practices should not only be applied to data messages but also to signaling messages — or it may lead to signal injection [Goo+11].

**KEY MANAGEMENT** In addition to those properties, key management should also be carefully considered. For instance, *re-keying* [Med+10] involves frequent changes of the key used for encrypted sessions. It is particularly suited for low-cost devices, such as RFID tags in the proposed example. Such a strategy prevents an attacker from collecting sufficient traces before a key update. While it does not prevent the side-channel itself in terms of information leakage, it makes the side-channel threat model impossible — *i.e.*, preventing the capture of a sufficiently high number of traces using a fixed key.

**ATTACKER CAPABILITIES** Another consideration is the encryption trigger capabilities for an attacker. For instance, a specification may systematically impose limits on: the number of connections in a fixed amount of time, or on the number of failed connection, or by adding an exponentially increasing time penalty for each failed connection attempt. As re-keying, it does not prevent the side-channel information leakage, but

it also significantly complicates the attack — *i.e.*, making hard to collect a high number of traces in a reasonable amount of time.

Specification countermeasures may be the simplest ones and the most efficient against side channels. Moreover, they can precisely target an attack, as for Screaming Channels in Part II — in which we conclude that a specification which would specify precisely that the radio should not be enabled during the computation of a [cryptographic operation](#) is a simple and valid countermeasure. However, while it may sound obvious to the reader, this will only work when considering side channels attacking protocols (defined in specifications) — and cannot prevent attacks such as the one presented in Part III leveraging the phase-modulated [compromising emanation](#). Introducing significant changes in a protocol specification may also be difficult in practice since it often requires the cooperation of several actors and may break retro-compatibility with previous versions depending on the considered countermeasure.

#### 10.1.4 Software

Countermeasures implemented in software are limited when considering physical attacks, but some examples have been proposed in the literature. Generally, the Soft-TEMPEST technique (introduced in Section 3.3.3), in particular signal generation, may be leveraged to jam or cancel [compromising emanation](#) by generating a stronger or out-of-phase signal, respectively. However, the limitations of signal generation techniques in terms of center frequency, bandwidth, and power may complicate its use. Moreover, it demands a deep knowledge about the actual [compromising emanation](#) — which as we saw previously, are often unexpected.

Another example of a software countermeasure against eavesdropping of text information from computer monitors through [compromising emanation](#) (presented in Section 3.3.4) is based on the use of special fonts. Kuhn and Anderson introduced *Tempest Fonts*, also known as *Safe Fonts* from Kubiak *et al.* [Kub19], that are designed to use only right angles and rectangular shapes, avoiding decorative elements. Using those simple rules, a human person that previously successfully read text using [compromising emanation](#) emitted by monitors or video interfaces (*e.g.*, VGA, DVI, HDMI) was unable to read it when using safer fonts. However, this kind of technique is obviously tightly coupled to a particular attack.

## 10.2 LESSONS LEARNED

In this last section, we propose to take a step back about the nature of those security issues and the way the hardware security community mitigates them.

**ROOT CAUSES** As seen through the state of the art (Part *i*), *BlueScream* (Part *ii*) or *PhaseSCA* (Part *iii*), there is a common cause between all those security issues. From a high-level perspective, one may conclude that the interaction between the software and the hardware – processing the secret information – with the physical environment is the root cause of the information leakage highlighted in this thesis. However, when inspecting those issues in depth, the phenomena studied in **electromagnetic compatibility** are the main root causes the security threats. Unfortunately, hardware security researchers may come from a computer science background, without a deep knowledge in **electromagnetic compatibility (EMC)**. Worse, electronic designers and **EMC** compliance tests do not assess the **electromagnetic radiation** from a security perspective, especially considering information leakage. As such, it is clear that those security issues will be complicated to mitigate without strong cooperation between those fields — and a wide interdisciplinary background for hardware security researchers.

**SOFTWARE SECURITY COMPARISON** A classical workflow for software security is to find a vulnerability, deploy a patch fixing it, then the problem is considered as solved. Considering protocol stacks security, manufacturers cannot completely rely on this mechanism, because patching embedded devices may be hard or even impossible — for embedded devices without connectivity. Moreover, their programmable memory is often limited, leading the manufacturers to arbitrate between different security patches (with one patch taking precedence over another). Considering hardware security, manufacturers cannot rely on this workflow at all, since patching the hardware of an embedded device is costly and imply to perform physical modifications or more often, replacing the device. It is common to fix hardware security issues by deploying a new hardware design in a new version of the device, thus leaving already deployed equipment vulnerable to attacks. Thus, in the real world, a significant amount of vulnerable devices are deployed in the wild and will never be patched.

**DESIGN CONSIDERATIONS** According to the claims above, we can conclude that protocols and hardware security must be addressed by design and not by adding patches to the design. As stated by the quote of this part, a good design taking security into account from the start removes the need to patch it afterwards. We does not claim that this is possible to have a perfectly secure device from the factory to its end of life in a single try, but consider that applying a set of good practices during the design phase may significantly improve the situation. From our perspective, the main principle that should be emphasized is the following:

The possibility to patch an implementation should be ensured by the design itself, but the design should not be patched — or the entire device will be flawed.

A parallel with micro-architectural security would be with transient execution attacks [CKG20], in particular the ones that exploit speculative execution. While a good design would have allowed to disable speculative execution for security-critical devices – at the cost of a significant performance decrease, to our knowledge, it is impossible to do so because it is part of the design. Regarding protocol specifications, the majority of countermeasures cited above are known from at least one decade, but are generally not implemented in widespread [Internet of Things](#) protocols. Good and secure designs, from the silicon to the protocol specification, should ideally take security into account systematically, since the possibilities offered by the countermeasures depicted before at the different layers may significantly reduce the associated security risks. While physical security issues are often difficult to anticipate, multiple countermeasures are well-known today and could drastically reduce the associated security risks.



THIS thesis opened up two promising research directions for the future. First, *BlueScream* demonstrates and evaluates how Screaming Channels can be a realistic threat to future [Internet of Things](#) communications. Second, *PhaseSCA* uncover how to exploit phase-modulated [compromising emanation](#) in side-channel attacks, paving the way for new attack strategies and highlighting the need for systemic countermeasures. This thesis presents our first steps exploring a whole new research thematic, as such, improvements and additional researches will be conducted in the next years. Reproducibility is fundamental for us, motivating us to open-source both our code and data for all of our projects — such that they can be reviewed, replicated, and extended.

In this section, we will first depict the current limitations of our work in Section [11.1](#). Then, we will expose the future research directions we plan to conduct to extend this work in Section [11.2](#). Finally, we will conclude our work in Section [11.3](#).

### 11.1 LIMITATIONS

**BLUESCREAM** During this project, we spent a lot of time to learn about research methodology, side-channel optimizations, radio communications, and [electromagnetic compatibility](#), since contributing to this field in a relevant way requires interdisciplinary skills. The investigation involved a lot of manual work at the beginning, such as research and extraction of the leakage from the spectrum, tuning of [Bluetooth Low Energy](#) parameters to evaluate their impacts, or tuning the side-channel attack parameters. While we automated a significant part of these steps to allow their reproducibility with minimal manual intervention, having an end-to-end replication of our work will still require a solid experience in the field. Moreover, the methodology that we leveraged to experiment with the [Bluetooth Low Energy](#) has room for improvements, discussed in Section [11.2](#). Additionally, collecting datasets of a real protocol instead of using instrumented firmware was also a significant challenge, because it involved a lot of custom instrumentation code and appropriate storage backend. With a stronger collaboration with device manufacturers instead of performing black-box testing, it should be possible to have of better understanding of how the leakage propagates and which hardware element is responsible. Finally, our leakage exploitation does not leverage some well-known available techniques

from radio communication to improve the attack efficiency, such as frequency diversity.

**PHASESCA** This project can be improved in several ways. First, the generalization of our results remains an open question, as this work uses multiple popular and COTS [system-on-chips](#) but remains not exhaustive. Second, the leakage source study is partial, in the sense that we explored the hypothesis that we had at that time — but “electrons do not read schematics” [Pau06, p. 763], and it is possible that other phenomena are causing jitters and therefore phase-modulated [compromising emanation](#) — instead of only [coupling](#) between digital activity and oscillators.

## 11.2 FUTURE WORK

### 11.2.1 *Continuing Our Work*

**BLUESCREAM** The first promising direction with this project is the generalization of the [Internet of Things](#) protocol assessment. Leveraging WHAD [CC24], the attacker-side stack implementation and framework for packet injection that we used in this project, it should be possible to create a general methodology to assess new protocols vulnerabilities. In particular, we imagine how a tool combining microbenchmarking – to isolate which operations is susceptible to leak – and protocol fuzzing – to isolate in which state a protocol is susceptible to leak – could lead to a generic tool. Combining this approach and our setup of simultaneous dual leakage detection in both [near-field \(NF\)](#) and [far-field \(FF\)](#), it should be possible to fully automate this process and efficiently discover how to “make a protocol scream”. However, exploring this direction would still imply a significant engineering effort. The second promising direction is the generalization of the attack to other vulnerable chipsets. Currently, the nRF52832 has been extensively studied by the security community. Some attempts tried to attack most challenging electronic devices, such as satellite communication systems [Gal24b; Gal24a; GJ24] or fully-digital radio devices [Gui24]. While a weak leakage has been detected in those systems, it was orders of magnitude smaller than the nRF52832. The exploitation of challenging electronic devices is not as easy as with [Internet of Things](#) devices, however, it is highly probable that other [Internet of Things system-on-chips](#) are vulnerable as well. The third promising research direction is to leverage the last side-channel optimizations from the state of the art. The technique discovered after this project, using a combination of amplitude-modulated and phase-modulated leakage, could be a promising approach to optimize this attack. Additionally, we could also perform frequency diversity and other techniques of side channel



optimization — such as intermediate values combination [MOW14] or performing a multi-dimensional attack [GGH19]. We strongly believe that by leveraging all previously described techniques, Screaming Channels could become a serious threat and that manufacturers and protocol consortiums should consider this in their threat model.

**PHASESCA** This project also leads to fruitful future work, with the first direction being the impact of the phase exploitation. For example, all other **electromagnetic** attacks (*e.g.*, Van Eck Phreaking) or **electromagnetic** side-channel attacks (*e.g.*, on hardware cryptographic block) do not exploit the potential phase-modulated leakage. This could improve significantly their performances and even enable attacks that were previously not strong enough to be considered as a threat. Moreover, an interesting comparison would be to evaluate the same side-channel attack and setup using an oscilloscope on one side and a **software-defined radio (SDR)** (exploiting both the amplitude and the phase) on the other side. This work may allow the community to determine which hardware is the best suited for which target. Finally, working on those projects made us realize that other important directions are waiting to be explored for side channel research in general.

### 11.2.2 Improving Side Channels

**ELECTROMAGNETIC SIDE CHANNELS** By carefully reviewing the literature of **electromagnetic** attacks in hardware security, we find that the use of **electromagnetic compatibility (EMC)** concepts is inconsistent between different papers, underlining definition and classification issues. Therefore, a systematic classification of **electromagnetic** attacks leveraging a state of the art **EMC** knowledge would be an important contribution. Concerning side-channel analysis, an interesting way of studying them is to consider and model them as communication channels. As such, an interesting venue would be to systematically analyze each **electromagnetic wave** parameters that can be modulated to communicate information. In this thesis, we focused our work on time-domain parameters (*e.g.*, amplitude and phase), but we did not consider spatial parameters (*e.g.*, polarization), which remains unexplored. Finally, **electromagnetic** analysis can be done in different domains. The standard way is to perform a time-domain analysis, but existing work leveraged the frequency-domain through Fast Fourier Transform (FFT) [Sch+10; MG10] or through Short-Time Fourier Transform (STFT) [VP09] (also known as waterfall or spectrogram). Recent work leveraged time-frequency domain through the Wavelet Transform (WT) [Des22], giving a novel vision of visualizing and exploiting side-channel signals.

**SIDE CHANNELS** In the side-channel attack community, we noticed that open-source code is not systematically provided when publishing new results. Moreover, every team seems to work with its own datasets and internal tools, significantly complicating comparison with prior works. As such, it would be a considerable advantage to have a more open ecosystem. First, by providing open-source code and reference datasets, in the same way that the ASCAD [ANS21] initiative from the French ANSSI.<sup>1</sup> Second, the development and adoption of a general, modular, and state-of-the-art side-channel framework would be a new step toward open research. The ideal scenario would be a framework allowing researchers to compare their side channels on shared datasets, to ensure result consistency, and develop their code in reusable modules, to ensure reproducibility and improve the wide adoption of new side-channel techniques. Such examples in other research areas are Mirage [Cay22] and then WHAD [CC24] for wireless security, or LibAFL [Fio+22] for fuzzing. The previous discussion will address reproducibility of *processing*, but not the reproducibility of the *measure*. The latter challenge does not have any simple answer, but the author wants to highlight two complementary approaches that may be followed by researchers to improve the reproducibility of measurement. The first one is about the rigor of documenting the experiments — the hardware (models and characteristics), the environment (schemes, pictures, measurements) — and providing the software (source code and binary firmware). The second one is about the creation of new, open, and online platforms dedicated to researchers to perform remote measurements — inspired from how computer scientists in high-performance computing are sharing their resources.

*Reproducibility of measurements and processing are key factors to improve the future research on side channels.*

### 11.3 CONCLUSION

Through this thesis, we have reviewed radio communication and electromagnetic compatibility knowledge to understand the state of the art of electromagnetic attacks. We hope to have shown that a better understanding of electromagnetic compatibility background would be beneficial for the hardware security community studying electromagnetic attacks.

Leveraging this knowledge, we proposed two main contributions. Our first contribution studies the applicability of the long-distance electromagnetic side-channel attack called Screaming Channels to modern Internet of Things protocols. Our result provide a methodology and evaluation of this threat on the Bluetooth Low Energy protocol, published as *BlueScream* [Ayo+24a]. With limited performance results and a significant amount of engineering work, we showed that the attack is

<sup>1</sup> We however regret that the ASCAD project is not enriched across time with new datasets.

feasible on the Bluetooth Low Energy protocol and that it is an emergent threat that should be addressed in the future.

Our second result consist in the uncovering of phase-modulated side-channel leakage in modern embedded devices. We developed a methodology and softwares to exploit this novel side-channel vector, published as *PhaseSCA* [Ayo+24b]. The outcome is a deeper understanding of how digital activity modulates electromagnetic leakage and how to use this understanding to improve electromagnetic attack performances. Moreover, this work allowed us to create a bridge between multiple publications in the literature that were not connected — while exploiting the same root cause phenomenon.

Based on the experience learned from those contributions, we were able to dress perspectives about compromising emanations in embedded devices. We provided an insight into which countermeasures and scenarios should be addressed for a good design regarding the electromagnetic side-channel attack threat model.

During all our work, we strive to make our code and data public for all of our projects, in an effort towards reproducible research. They are hosted on well-known platform (GitHub and Zenodo) and duplicated to prevent data loss.

By opening new research directions and by bridging multiple disciplines together, we hope that this work will be beneficial to researchers, engineers and students who are interested in electromagnetic attacks and information leakages through unintentional compromising emanations.



## BIBLIOGRAPHY

- [IEC90] International Electrotechnical Commission (IEC). (*Electromagnetic Radiation (161-01-10)*). 1990. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=161-01-10> (cit. on p. 21).
- [IEC92] International Electrotechnical Commission (IEC). (*Crosstalk (722-15-03)*). 1992. URL: <https://electropedia.org/iev/iev.nsf/display?openform&ievref=722-15-03> (cit. on p. 33).
- [IEC02] International Electrotechnical Commission (IEC). (*Non-Linear (131-11-19)*). 2002. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=131-11-19> (cit. on p. 30).
- [IEC17] International Electrotechnical Commission (IEC). (*Intermodulation (161-06-20)*). 2017. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=161-06-20> (cit. on p. 32).
- [IEC18a] International Electrotechnical Commission (IEC). (*Cross-Modulation (161-06-19)*). 2018. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=161-06-19> (cit. on p. 32).
- [IEC18b] International Electrotechnical Commission (IEC). (*Electromagnetic Compatibility (161-01-07)*). 2018. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=161-01-07> (cit. on p. 19).
- [IEC18c] International Electrotechnical Commission (IEC). (*Electromagnetic Disturbance (161-01-05)*). 2018. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=161-01-05> (cit. on p. 20).
- [IEC18d] International Electrotechnical Commission (IEC). (*Electromagnetic Interference (161-01-06)*). 2018. URL: <https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=161-01-06> (cit. on p. 20).
- [Age72] National Security Agency. *TEMPEST: A Signal Problem*. Tech. rep. NSA, 1972 (cit. on pp. 5, 46).
- [Agr+03] Dakshi Agrawal, Bruce Archambeault, Josyula Rao, and Pankaj Rohatgi. *The EM Side-Channel(s): Attacks and Assessment Methodologies*. Tech. rep. IBM, 2003 (cit. on pp. 41, 42, 117, 118, 129, 145).

- [Agr+02] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. "The EM Side-Channel(s)." In: *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*. CHES '02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 29–45. ISBN: 3540004092 (cit. on pp. 42, 60).
- [ARR03] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. "Multi-Channel Attacks." In: *Workshop on Cryptographic Hardware and Embedded Systems*. 2003. URL: <https://api.semanticscholar.org/CorpusID:45546568> (cit. on p. 59).
- [Air98a] Secretary of the Air Force. *Air Force Systems Security INSTRUCTION (AFSSI) 7010: Emission Security Assessments*. Tech. rep. US Air Force, May 1998. URL: <https://cryptome.org/afssi-7010.htm> (cit. on pp. 47, 48).
- [Air98b] Secretary of the Air Force. *Air Force Systems Security Memorandum (AFSSM) 7011: Emission Security Countermeasures Review*. Tech. rep. US Air Force, May 1998. URL: <https://cryptome.org/afssm-7011.htm> (cit. on pp. 47, 48).
- [AS09] Charles K. Alexander and Matthew N.O. Sadiku. *Fundamentals of Electric Circuits*. Ed. by McGraw-Hill. 4th Edition. 2009. ISBN: 978-0-07-352955-4 (cit. on p. xxii).
- [Ali+22] Abubakar S. Ali, Michael Baddeley, Lina Bariah, Martin Andreoni Lopez, Willian Tessaro Lunardi, Jean-Pierre Giacalone, and Sami Muhaidat. "JamRF: Performance Analysis, Evaluation, and Implementation of RF Jamming Over Wi-Fi." In: *IEEE Access* 10 (2022), pp. 133370–133384. doi: 10.1109/ACCESS.2022.3230895 (cit. on p. 53).
- [Ami+07] Frederic Amiel, Karine Villegas, Benoit Feix, and Louis Marcel. "Passive and Active Combined Attacks: Combining Fault Attacks and Side Channel Analysis." In: *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC 2007)*. 2007, pp. 92–102. doi: 10.1109/FDTC.2007.12 (cit. on p. 54).
- [AK99] Ross J. Anderson and M. Kuhn. "Soft Tempest - An Opportunity for NATO." In: *undefined* (1999). URL: <https://www.semanticscholar.org/paper/Soft-Tempest-%7B-An-Opportunity-for-NATO-Anderson-Kuhn/39f9fa0a49958dd552c76693bc3b9647> (visited on 05/11/2022) (cit. on pp. 6, 46–48, 50).
- [ANS21] ANSSI. *ASCAD: ANSSI SCA Database*. 2021. URL: <https://github.com/ANSSI-FR/ASCAD> (cit. on p. 162).

- [Ant23] Daniele Antonioli. “BLUFFS: Bluetooth Forward and Future Secrecy Attacks and Defenses.” In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’23. New York, NY, USA: Association for Computing Machinery, 2023, pp. 636–650. ISBN: 9798400700507. DOI: [10.1145/3576915.3623066](https://doi.org/10.1145/3576915.3623066). URL: <https://doi.org/10.1145/3576915.3623066> (cit. on p. 80).
- [Ayo24] Pierre Ayoub. *Screaming Channels on Bluetooth Low Energy*. ACSAC24 Artifact Evaluation. Zenodo, Aug. 2024. DOI: [10.5281/zenodo.13384278](https://doi.org/10.5281/zenodo.13384278). URL: <https://zenodo.org/records/13384278> (cit. on p. 71).
- [Ayo25a] Pierre Ayoub. *PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels - ATmega328 Dataset*. Zenodo, Feb. 2025. DOI: [10.5281/zenodo.14800719](https://doi.org/10.5281/zenodo.14800719). URL: <https://doi.org/10.5281/zenodo.14800719> (cit. on p. 114).
- [Ayo25b] Pierre Ayoub. *PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels - nRF51422 Dataset*. Zenodo, Feb. 2025. DOI: [10.5281/zenodo.14849086](https://doi.org/10.5281/zenodo.14849086). URL: <https://doi.org/10.5281/zenodo.14849086> (cit. on p. 114).
- [Ayo25c] Pierre Ayoub. *PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels - nRF52832 Dataset*. Zenodo, Feb. 2025. DOI: [10.5281/zenodo.14800633](https://doi.org/10.5281/zenodo.14800633). URL: <https://doi.org/10.5281/zenodo.14800633> (cit. on p. 114).
- [Ayo25d] Pierre Ayoub. *PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels - STM32L1 Dataset*. Zenodo, Feb. 2025. DOI: [10.5281/zenodo.14800774](https://doi.org/10.5281/zenodo.14800774). URL: <https://doi.org/10.5281/zenodo.14800774> (cit. on p. 114).
- [Ayo+24a] Pierre Ayoub, Romain Cayre, Aurélien Francillon, and Clémentine Maurice. “BlueScream: Screaming Channels on Bluetooth Low Energy.” In: *40th Annual Computer Security Applications Conference (ACSAC ’24)*. Waikiki, Honolulu, Hawaii, United States, Dec. 2024. URL: <https://hal.science/hal-04725668> (cit. on pp. 69, 162).
- [Ayo+24b] Pierre Ayoub, Aurélien Hernandez, Romain Cayre, Aurélien Francillon, and Clémentine Maurice. “PhaseSCA: Exploiting Phase-Modulated Emanations in Side Channels.” In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) 2025.1* (Dec. 2024), pp. 392–419. DOI: [10.46586/tches.v2025.i1.392-419](https://doi.org/10.46586/tches.v2025.i1.392-419). URL: <https://tches.iacr.org/index.php/TCHES/article/view/11934> (cit. on pp. 109, 163).

- [Bac+09] Michael Backes, Tongbo Chen, Markus Duermuth, Hendrik P.A. Lensch, and Martin Welk. “Tempest in a Teapot: Compromising Reflections Revisited.” In: *2009 30th IEEE Symposium on Security and Privacy*. 2009, pp. 315–327. DOI: [10.1109/SP.2009.20](https://doi.org/10.1109/SP.2009.20) (cit. on p. 56).
- [Bal+19] E. Balestrieri, F. Picariello, S. Rapuano, and I. Tudosa. “Review on Jitter Terminology and Definitions.” In: *Measurement* 145 (2019), pp. 264–273. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2019.05.047>. URL: <https://www.sciencedirect.com/science/article/pii/S0263224119304737> (cit. on p. 115).
- [BB13] Andrea Barisani and Daniele Bianco. “Sniffing Keystrokes with Lasers and Voltmeters.” In: *DEFCON’17*. 2013. URL: <https://www.youtube.com/watch?v=Amnv4ncqKtA> (cit. on p. 50).
- [Bar+02] N. Barton, D. Ozis, T. Fiez, and K. Mayaram. “The Effect of Supply and Substrate Noise on Jitter in Ring Oscillators.” In: *Proceedings of the IEEE 2002 Custom Integrated Circuits Conference (Cat. No.02CH37285)*. 2002, pp. 505–508. DOI: [10.1109/CICC.2002.1012890](https://doi.org/10.1109/CICC.2002.1012890) (cit. on p. 129).
- [Beh07] Arya Behzad. *Wireless LAN Radios*. Wiley, May 2007. ISBN: 9780470209301. DOI: [10.1002/9780470209301](https://doi.org/10.1002/9780470209301). URL: <http://dx.doi.org/10.1002/9780470209301> (cit. on p. 18).
- [Ber05] Daniel J. Bernstein. “Cache-Timing Attacks on AES.” In: 2005. URL: <https://api.semanticscholar.org/CorpusID:2217245> (cit. on pp. 6, 34).
- [BS97] Eli Biham and Adi Shamir. “Differential fault analysis of secret key cryptosystems.” In: *Advances in Cryptology — CRYPTO ’97*. Ed. by Burton S. Kaliski. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 513–525. ISBN: 978-3-540-69528-8. DOI: [10.1007/BFb0052259](https://doi.org/10.1007/BFb0052259) (cit. on p. 54).
- [Blo18] Android Developers Blog. *Discontinuing support for Android Nearby Notifications*. 2018. URL: <https://android-developers.googleblog.com/2018/10/discontinuing-support-for-android.html> (cit. on pp. 70, 76).
- [BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. “Correlation Power Analysis with a Leakage Model.” In: *Cryptographic Hardware and Embedded Systems - CHES 2004*. Ed. by Marc Joye and Jean-Jacques Quisquater. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29. ISBN: 978-3-540-28632-5 (cit. on pp. 57, 126).



- [Buh+22] Ileana Buhan, Lejla Batina, Yuval Yarom, and Patrick Schaulmont. “SoK: Design Tools for Side-Channel-Aware Implementations.” In: *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. ASIA CCS ’22. Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 756–770. ISBN: 9781450391405. DOI: [10.1145/3488932.3517415](https://doi.org/10.1145/3488932.3517415). URL: <https://doi.org/10.1145/3488932.3517415> (cit. on p. 64).
- [BEA08] Andrew Burnside, Ahmet Erdogan, and Tughrul Arslan. “The Re-Emission Side Channel.” In: *2008 Bio-inspired, Learning and Intelligent Systems for Security*. Aug. 2008, pp. 154–159. DOI: [10.1109/BLISS.2008.22](https://doi.org/10.1109/BLISS.2008.22) (cit. on pp. 48–50).
- [CZP14] Robert Callan, Alenka Zajić, and Milos Prvulovic. “A Practical Methodology for Measuring the Side-Channel Signal Available to the Attacker for Instruction-Level Events.” In: *Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO-47. USA: IEEE Computer Society, Dec. 2014, pp. 242–254. ISBN: 978-1-4799-6998-2. DOI: [10.1109/MICRO.2014.39](https://doi.org/10.1109/MICRO.2014.39). URL: <https://doi.org/10.1109/MICRO.2014.39> (visited on 10/23/2021) (cit. on pp. 50, 64).
- [CZP15] Robert Callan, Alenka Zajić, and Milos Prvulovic. “FASE: Finding Amplitude-Modulated Side-Channel Emanations.” In: *Proceedings of the 42nd Annual International Symposium on Computer Architecture*. ISCA ’15. New York, NY, USA: Association for Computing Machinery, June 2015, pp. 592–603. ISBN: 978-1-4503-3402-0. DOI: [10.1145/2749469.2750394](https://doi.org/10.1145/2749469.2750394). URL: <https://doi.org/10.1145/2749469.2750394> (visited on 10/23/2021) (cit. on pp. 50, 64).
- [Cam20] Giovanni Camurati. “Security Threats Emerging from the Interaction Between Digital Activity and Radio Transceiver.” PhD thesis. Sorbonne Université, Dec. 2020. URL: <https://theses.hal.science/tel-03414339> (cit. on p. 61).
- [CDS22] Giovanni Camurati, Matteo Dell’Amico, and François-Xavier Standaert. “MCRank: Monte Carlo Key Rank Estimation for Side-Channel Security Evaluations.” In: *IACR Transactions on Cryptographic Hardware and Embedded Systems 2023*, Issue 1 (2022), pp. 277–300. DOI: [10.46586/tches.v2023.i1.277-300](https://doi.org/10.46586/tches.v2023.i1.277-300). URL: <https://tches.iacr.org/index.php/TCHES/article/view/9953> (cit. on p. 60).
- [CF22] Giovanni Camurati and Aurélien Francillon. “Noise-SDR: Arbitrary Modulation of Electromagnetic Noise from Unprivileged Software and Its Impact on Emission Secu-

- rity." In: *2022 IEEE Symposium on Security and Privacy (S&P)*. 2022, pp. 1193–1210. DOI: [10.1109/SP46214.2022.9833767](https://doi.org/10.1109/SP46214.2022.9833767) (cit. on pp. 6, 44, 51).
- [CFS20] Giovanni Camurati, Aurélien Francillon, and François-Xavier Standaert. "Understanding Screaming Channels: From a Detailed Analysis to Improved Attacks." In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (CHES 2020)* 2020.3 (June 2020), pp. 358–401. DOI: [10.13154/tches.v2020.i3.358-401](https://doi.org/10.13154/tches.v2020.i3.358-401). URL: <https://tches.iacr.org/index.php/TCHES/article/view/8594> (cit. on pp. 6, 35, 61, 62, 70, 76, 80, 90, 94, 126).
- [Cam+18] Giovanni Camurati, Sebastian Poehlau, Marius Muench, Tom Hayes, and Aurélien Francillon. "Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers." In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 163–177. ISBN: 9781450356930. DOI: [10.1145/3243734.3243802](https://doi.org/10.1145/3243734.3243802). URL: <https://doi.org/10.1145/3243734.3243802> (cit. on pp. xxiv, 6, 7, 42, 61, 62, 69, 76, 85, 86, 91, 94, 100, 109, 116, 119, 126).
- [CKG20] Claudio Canella, Khaled N. Khasawneh, and Daniel Gruss. "The Evolution of Transient-Execution Attacks." In: *Proceedings of the 2020 on Great Lakes Symposium on VLSI*. GLSVLSI '20. Virtual Event, China: Association for Computing Machinery, 2020, pp. 163–168. ISBN: 9781450379441. DOI: [10.1145/3386263.3407583](https://doi.org/10.1145/3386263.3407583). URL: <https://doi.org/10.1145/3386263.3407583> (cit. on p. 157).
- [Cao+23] Pei Cao, Chi Zhang, Xiang-Jun Lu, Hai-Ning Lu, and Da-Wu Gu. "Side-Channel Analysis for the Re-Keying Protocol of Bluetooth Low Energy." In: *Journal of Computer Science and Technology* 38.5 (Sept. 2023), pp. 1132–1148. ISSN: 1860-4749. DOI: [10.1007/s11390-022-1229-3](https://doi.org/10.1007/s11390-022-1229-3). URL: <http://dx.doi.org/10.1007/s11390-022-1229-3> (cit. on pp. 76–78, 93, 106).
- [CC24] Damien Cauquil and Romain Cayre. *One For All and All For WHAD: Wireless Shenanigans Made Easy!* DEF CON 2024, DEF CON Security Conference, 8-11 August 2024, Las Vegas, NV, USA. 2024. URL: <https://defcon.org/html/defcon-32/dc-32-speakers.html> (cit. on pp. 78, 84, 160, 162).
- [Cay22] Romain Cayre. "Offensive and Defensive Approaches for Wireless Communication Protocols Security in IoT." PhD thesis. INSA de Toulouse, June 2022. URL: <https://laas.hal.science/tel-03841305> (cit. on p. 162).

- [Cay24] Romain Cayre. *ButteRFly*. 2024. URL: <https://github.com/RCayre/injectable-firmware> (cit. on p. 84).
- [Cay+21a] Romain Cayre, Florent Galtier, Guillaume Auriol, Vincent Nicomette, Mohamed Kaâniche, and Géraldine Marconato. “InjectaBLE: Injecting malicious traffic into established Bluetooth Low Energy connections.” In: *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2021)*. Taipei (virtual), Taiwan, June 2021. DOI: [10.1109/DSN48987.2021.00050](https://doi.org/10.1109/DSN48987.2021.00050). URL: <https://laas.hal.science/hal-03193297> (cit. on p. 84).
- [Cay+21b] Romain Cayre, Florent Galtier, Guillaume Auriol, Vincent Nicomette, Mohamed Kaâniche, and Géraldine Marconato. “WazaBee: Attacking Zigbee Networks by Diverting Bluetooth Low Energy Chips.” In: *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2021)*. Taipei (virtual), Taiwan, June 2021. DOI: [10.1109/DSN48987.2021.00049](https://doi.org/10.1109/DSN48987.2021.00049). URL: <https://laas.hal.science/hal-03193299> (cit. on p. 6).
- [CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. “Template Attacks.” In: *Workshop on Cryptographic Hardware and Embedded Systems*. 2002. URL: <https://api.semanticscholar.org/CorpusID:9694193> (cit. on pp. 58, 99).
- [CYC20] Jieun Choi, Hae-Yong Yang, and Dong-Ho Cho. “TEMPEST Comeback: A Realistic Audio Eavesdropping Threat on Mixed-Signal SoCs.” In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. Virtual Event USA: ACM, Oct. 2020, pp. 1085–1101. ISBN: 978-1-4503-7089-9. DOI: [10.1145/3372297.3417241](https://doi.org/10.1145/3372297.3417241). URL: <https://dl.acm.org/doi/10.1145/3372297.3417241> (visited on 11/10/2020) (cit. on pp. 6, 7, 41, 43).
- [Col+18] Travis F. Collins, Robin Getz, Di Pu, and Alexander M. Wyglinski. *Software-Defined Radio for Engineers*. Ed. by Artech House. 2018. ISBN: 9781630814595. URL: <https://ieeexplore.ieee.org/document/9100100> (cit. on pp. 3, 17, 18).
- [Com14] Douglas E. Comer. *Internetworking with TCP/IP. Principles, Protocol, and Architecture*. Ed. by Pearson. 6th Edition. Vol. 1. 2014. ISBN: 978-0-13-608530-0 (cit. on p. 3).
- [Cui15] Ang Cui. *Emanate Like a Boss: Generalized Covert Data Exfiltration with Funtenna*. Black Hat USA 2015. 2015. URL: <http://www.funtenna.org/> (cit. on p. 51).

- [DS18] Nicola Da Dalt and Ali Sheikholeslami. *Understanding Jitter and Phase Noise: A Circuits and Systems Perspective*. 1st ed. Cambridge University Press, Feb. 22, 2018. ISBN: 978-1-107-18857-0 978-1-316-98123-8. DOI: [10.1017/9781316981238](https://doi.org/10.1017/9781316981238). URL: <https://www.cambridge.org/core/product/identifier/9781316981238/type/book> (visited on 07/15/2024) (cit. on p. 115).
- [Dan+24] Erez Danieli, Menachem Goldzweig, Moshe Avital, and Itamar Levi. "Revealing the Secrets of Radio Embedded Systems: Extraction of Raw Information via RF." In: *IEEE Transactions on Information Forensics and Security* 19 (2024), pp. 2066–2081. ISSN: 1556-6021. DOI: [10.1109/tifs.2023.3345131](https://doi.org/10.1109/tifs.2023.3345131). URL: <http://dx.doi.org/10.1109/TIFS.2023.3345131> (cit. on pp. 6, 7, 63, 154).
- [Der91] Thomas J. Dermott. *NSA/CSS Regulation: Reg. No. 90-6*. Tech. rep. National Security Agency (NSA), 1991. URL: <https://cryptome.org/nsa-reg90-6.htm> (cit. on p. 48).
- [Des22] Gabriel Destouet. "Ondelettes pour le traitement des signaux compromettants." fr. PhD thesis. 2022. DOI: [10.34894/VQ1DJA](https://doi.org/10.34894/VQ1DJA). URL: <https://hal.science/tel-03758771> (cit. on p. 161).
- [Devo8a] Analog Devices. *Introduction to Electromagnetic Compatibility (EMC): EMI, RFI, and Shielding Concepts*. Tech. rep. Analog Devices, 2008. URL: <https://www.analog.com/media/en/training-seminars/tutorials/MT-095.pdf> (cit. on pp. 26, 34, 152).
- [Devo8b] Analog Devices. *Mixers and Modulators*. Tech. rep. Analog Devices, 2008. URL: <https://www.analog.com/media/en/training-seminars/tutorials/MT-080.pdf> (cit. on p. 30).
- [Did14] Les Histoires de Didymus, ed. *Bordeaux, 1834-36: Les frères Blanc craquent les télégraphes Chappe*. 2014. URL: <https://leshistoiresdedidymus.wordpress.com/2014/05/27/bordeaux-1834-36-les-freres-blanc-telegraphes-chappe/> (cit. on p. 1).
- [EG12] M. Abdelaziz Elaabid and Sylvain Guilley. "Portability of Templates." In: *Journal of Cryptographic Engineering* 2.1 (May 2012), pp. 63–74. DOI: [10.1007/s13389-012-0030-6](https://doi.org/10.1007/s13389-012-0030-6) (cit. on pp. 97, 99).
- [Est23] José Lopes Esteves. "Electromagnetic Interference and Information Security: Characterization, Exploitation and Forensic Analysis." PhD thesis. HESAM Université, ANSSI, CNAM, June 2023. URL: <https://theses.hal.science/tel-04155509> (cit. on p. 45).

- [Fan+22] Clément Fanjas, Clément Gainé, Driss Aboulkassimi, Simon Pontié, and Olivier Potin. “Combined Fault Injection and Real-Time Side-Channel Analysis for Android Secure-Boot Bypassing.” In: *Smart Card Research and Advanced Applications - 21st International Conference, CARDIS 2022, Birmingham, UK, November 7-9, 2022, Revised Selected Papers*. 2022, pp. 25–44. DOI: [10.1007/978-3-031-25319-5\\_2](https://doi.org/10.1007/978-3-031-25319-5_2). URL: [https://doi.org/10.1007/978-3-031-25319-5\\_2](https://doi.org/10.1007/978-3-031-25319-5_2) (cit. on p. 63).
- [Fen+23] Justin Feng, Timothy Jacques, Omid Abari, and Nader Sehatbakhsh. “Everything has its Bad Side and Good Side: Turning Processors to Low Overhead Radios Using Side-Channels.” In: *The 22nd International Conference on Information Processing in Sensor Networks* (May 2023). DOI: [10.1145/3583120.3586959](https://doi.org/10.1145/3583120.3586959). URL: <http://dx.doi.org/10.1145/3583120.3586959> (cit. on p. 51).
- [Fer+24] Santiago Fernández, Emilio Martínez, Gabriel Varela, Pablo Musé, and Federico Larroca. “Deep-TEMPEST: Using Deep Learning to Eavesdrop on HDMI from its Unintended Electromagnetic Emanations.” 2024. arXiv: [2407.09717](https://arxiv.org/abs/2407.09717) [cs.CR]. URL: <https://arxiv.org/abs/2407.09717> (cit. on p. 52).
- [FH08] Julie Ferrigno and Martin Hlaváč. “When AES Blinks: Introducing Optical Side Channel.” In: *IET Inf. Secur.* 2.3 (2008), pp. 94–98. DOI: [10.1049/iet-ifs:20080038](https://doi.org/10.1049/iet-ifs:20080038). URL: <https://doi.org/10.1049/iet-ifs:20080038> (cit. on pp. 5, 56).
- [Fig22] Louis Figuier. *Télégraphe Chappe*. 2022. URL: [https://commons.wikimedia.org/wiki/File:T%C3%A9l%C3%A9graphe\\_Chappe\\_2.jpg](https://commons.wikimedia.org/wiki/File:T%C3%A9l%C3%A9graphe_Chappe_2.jpg) (cit. on p. 2).
- [Fio+22] Andrea Fioraldi, Dominik Maier, Dongjia Zhang, and Davide Balzarotti. “LibAFL: A Framework to Build Modular and Reusable Fuzzers.” In: *Proceedings of the 29th ACM conference on Computer and communications security (CCS)*. CCS ’22. Los Angeles, U.S.A.: ACM, Nov. 2022 (cit. on p. 162).
- [Fou24a] Apache Software Foundation. *Mynewt*. 2024. URL: <https://mynewt.apache.org/> (cit. on p. 76).
- [Fou24b] Apache Software Foundation. *NimBLE*. 2024. URL: <https://github.com/apache/mynewt-nimble> (cit. on pp. 76, 85).
- [Fre16] Louis E. (Jr.) Frenzel. *Principles of Electronic Communication Systems*. Ed. by McGraw-Hill Education. 4th. 2016. ISBN: 9780073373850 (cit. on pp. 2, 3, 15, 24, 30, 32, 127).

- [Gal24a] Tom Gallagher. “Identifying System-on-a-Chip Data Leaks over Radio Transmissions of Small Satellites.” In: *International Journal of Applied Technology & Leadership (IJATL)* Volume 3 (Issue 1 Jan. 2024). ISSN: 2720-5215 (cit. on pp. 63, 160).
- [Gal24b] Tom Gallagher. “Understanding How System-on-a-Chip Data Can Leak over Radio Transmissions.” In: *International Journal of Applied Technology & Leadership (IJATL)* Volume 3 (Issue 1 Jan. 2024). ISSN: 2720-5215 (cit. on pp. 63, 160).
- [GJ24] Tom Gallagher and Michael Johnson. “Measuring System-on-a-Chip Data Leaks over Radio Transmissions of Small Satellites.” In: *International Journal of Applied Technology & Leadership (IJATL)* Volume 3 (Issue 1 Jan. 2024). ISSN: 2720-5215 (cit. on pp. 63, 160).
- [GMO01] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. “Electromagnetic Analysis: Concrete Results.” In: *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*. Vol. 2162. Lecture Notes in Computer Science. Springer, 2001, pp. 251–261. DOI: [10.1007/3-540-44709-1\\_21](https://doi.org/10.1007/3-540-44709-1_21) (cit. on p. 41).
- [GGH19] Christophe Genevey-Metat, Benoît Gérard, and Annelie Heuser. “Combining Sources of Side-Channel Information.” In: *C&ESAR*. 2019. URL: <https://api.semanticscholar.org/CorpusID:212783100> (cit. on pp. 59, 161).
- [Gen+22] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. “Stealing Keys from PCs Using a Radio: Cheap Electromagnetic Attacks on Windowed Exponentiation.” In: *Cryptographic Hardware and Embedded Systems – CHES 2015*. Berlin, Heidelberg: Springer-Verlag, 2022, pp. 207–228. ISBN: 978-3-662-48323-7. DOI: [10.1007/978-3-662-48324-4\\_11](https://doi.org/10.1007/978-3-662-48324-4_11). URL: <https://www.tau.ac.il/~tromer/radioexp/> (cit. on pp. 116, 119).
- [GST17] Daniel Genkin, Adi Shamir, and Eran Tromer. “Acoustic Cryptanalysis.” In: *J. Cryptology* 30 (2017), pp. 392–443. DOI: [10.1007/s00145-015-9224-2](https://doi.org/10.1007/s00145-015-9224-2) (cit. on pp. 5, 55).
- [GR20] Ilias Giechaskiel and Kasper Rasmussen. “Taxonomy and Challenges of Out-of-Band Signal Injection Attacks and Defenses.” In: *IEEE Communications Surveys & Tutorials* 22.1 (2020), pp. 645–670. ISSN: 2373-745X. DOI: [10.1109/comst.2019.2952858](https://doi.org/10.1109/comst.2019.2952858). URL: <http://dx.doi.org/10.1109/COMST.2019.2952858> (cit. on pp. 6, 153).



- [GSH20] D.V. Giri, Frank Sabath, and Richard Hoad. *High-Power Electromagnetic Effects on Electronic Systems*. Ed. by Artech House. 2020. ISBN: 978-1-63081-588-2. URL: <https://ieeexplore.ieee.org/document/9102227> (visited on 06/28/2021) (cit. on pp. 26–28, 49).
- [Glo+15] Cezary Glowacz, Vincent Grosso, Romain Poussier, Joachim Schüth, and François-Xavier Standaert. “Simpler and More Efficient Rank Estimation for Side-Channel Security Assessment.” In: *Fast Software Encryption*. Ed. by Gregor Leander. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 117–129. ISBN: 978-3-662-48116-5 (cit. on p. 60).
- [GKT19] Dennis R. E. Gnad, Jonas Krautter, and Mehdi B. Tahoori. “Leaky Noise: New Side-Channel Attack Vectors in Mixed-Signal IoT Devices.” In: *IACR Transactions on Cryptographic Hardware and Embedded Systems 2019.3* (May 2019), pp. 305–339. DOI: 10.13154/tches.v2019.i3.305-339. URL: <https://tches.iacr.org/index.php/TCHES/article/view/8297> (cit. on p. 7).
- [Goo+11] Travis Goodspeed, Sergey Bratus, Ricky Melgares, Rebecca Shapiro, and Ryan Speers. “Packets in Packets: Orson Welles’ In-Band Signaling Attacks for Modern Radios.” In: *5th USENIX Workshop on Offensive Technologies (WOOT 11)*. San Francisco, CA: USENIX Association, Aug. 2011. URL: <https://www.usenix.org/conference/woot11/packets-packets-orson-welles-band-signaling-attacks-modern-radios> (cit. on p. 154).
- [Goo15] Google. *Eddystone*. 2015. URL: <https://github.com/google/eddystone> (cit. on pp. 70, 76).
- [GoT24] GoTronic. *2,4 GHz Antenna*. 2024. URL: <https://www.go-tronic.fr/art-antenne-2-4-ghz-a24-hasm450-31786.htm> (cit. on p. 83).
- [Gou05] Louis Goubin. *Cartes à puce*. 2005. URL: <http://goubin.fr/papers/carte-a-puce.pdf> (cit. on p. 54).
- [Gra+21] Joseph Gravellier, Jean-Max Dutertre, Yannick Teglia, and Philippe Loubet Moundi. “SideLine: How Delay-Lines (May) Leak Secrets from Your SoC.” In: *Constructive Side-Channel Analysis and Secure Design*. Ed. by Shivam Bhasin and Fabrizio De Santis. Cham: Springer International Publishing, 2021, pp. 3–30. ISBN: 978-3-030-89915-8 (cit. on pp. 117, 119, 130).
- [Gra13] Eugene Grayver. *Implementing Software Defined Radio*. Springer New York, 2013. ISBN: 9781441993328. DOI: 10.1007/978-1-4419-9332-8. URL: <http://dx.doi.org/10.1007/978-1-4419-9332-8> (cit. on pp. xxiv, 3, 18).

- [Gro] Bluetooth Working Group. *Bluetooth Core Specification*. URL: <https://www.bluetooth.com/specifications/specs/core-specification-5-3/> (visited on 03/03/2023) (cit. on pp. 71–74, 87, 89, 90).
- [Gui24] Jeremy Guillaume. “Optimizing Data Leakage Exploitation in the Context of Screaming-Channel Attacks.” PhD thesis. CentraleSupélec, 2024 (cit. on pp. 63, 160).
- [Gui+24] Jeremy Guillaume, Maxime Pelcat, Amor Nafkha, and Rubén Salvador. “Attacking at Non-Harmonic Frequencies in Screaming-Channel Attacks.” In: *Smart Card Research and Advanced Applications*. Ed. by Shivam Bhasin and Thomas Roche. Cham: Springer Nature Switzerland, 2024, pp. 87–106. ISBN: 978-3-031-54409-5 (cit. on pp. 63, 100, 101).
- [Gui+22] Jeremy Guillaume, Maxime Pelcat, Amor Nafkha, and Ruben Salvador. “Virtual Triggering: a Technique to Segment Cryptographic Processes in Side-Channel Traces.” In: *2022 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, Nov. 2022. DOI: 10.1109/sips55645.2022.9919238. URL: <http://dx.doi.org/10.1109/SiPS55645.2022.9919238> (cit. on p. 62).
- [GME16] M. Guri, M. Monitz, and Y. Elovici. “USBee: Air-Gap Covert-Channel Via Electromagnetic Emission From USB.” In: *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. Dec. 2016, pp. 264–268. DOI: 10.1109/PST.2016.7906972. URL: <https://ieeexplore.ieee.org/document/7906972> (cit. on pp. 6, 44, 51).
- [Gur23] Mordechai Guri. “AIR-FI: Leaking Data From Air-Gapped Computers Using Wi-Fi Frequencies.” In: *IEEE Transactions on Dependable and Secure Computing* 20.3 (2023), pp. 2547–2564. DOI: 10.1109/TDSC.2022.3186627. URL: <https://ieeexplore.ieee.org/document/9808153> (cit. on pp. 6, 44, 51).
- [Gur+15] Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval Elovici. “GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies.” In: *24th USENIX Security Symposium (USENIX Security 15)*. 2015, pp. 849–864. ISBN: 978-1-939133-11-3. URL: <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/guri> (visited on 11/24/2020) (cit. on pp. 6, 44, 51).
- [Gur+14] Mordechai Guri, Gabi Kedma, Assaf Kachlon, and Yuval Elovici. “AirHopper: Bridging the Air-Gap between Isolated Networks and Mobile Phones Using Radio Fre-



- quencies." In: *arXiv:1411.0237 [cs]* (Nov. 2014). arXiv: [1411.0237 \[cs\]](https://arxiv.org/abs/1411.0237). URL: <http://arxiv.org/abs/1411.0237> (visited on 11/24/2020) (cit. on p. 44).
- [Hano4] Johnnie Hancock. "Jitter—Understanding It, Measuring It, Eliminating It Part 1: Jitter Fundamentals." In: (2004) (cit. on p. 115).
- [Han99] John W. Handy. *EI TEMPEST Installation Handbook (Air Force Qualification Training Package 2EXXX-202D)*. Tech. rep. Department of the Air Force, 1999. URL: [https://cdn.preterhuman.net/texts/government\\_information/intelligence\\_and\\_espionage/homebrew.military.and.espionage.electronics/servv89pn0aj.sn.sourcedns.com/\\_gbpprorg/mil/vaneck/nsa/HB202D.PDF](https://cdn.preterhuman.net/texts/government_information/intelligence_and_espionage/homebrew.military.and.espionage.electronics/servv89pn0aj.sn.sourcedns.com/_gbpprorg/mil/vaneck/nsa/HB202D.PDF) (cit. on p. 46).
- [HP00] P. Heydari and M. Pedram. "Analysis of Jitter due to Power-supply Noise in Phase-Locked Loops." In: *Proceedings of the IEEE 2000 Custom Integrated Circuits Conference (Cat. No.00CH37044)*. 2000, pp. 443–446. DOI: [10.1109/CICC.2000.852704](https://doi.org/10.1109/CICC.2000.852704) (cit. on p. 129).
- [HP01] P. Heydari and M. Pedram. "Jitter-Induced Power/Ground Noise in CMOS PLLs: a Design Perspective." In: *Proceedings 2001 IEEE International Conference on Computer Design: VLSI in Computers and Processors. ICCD 2001*. 2001 International Conference on Computer Design. ICCD 2001. Austin, TX, USA: IEEE Comput. Soc, 2001, pp. 209–213. ISBN: 978-0-7695-1200-6. DOI: [10.1109/ICCD.2001.955030](https://doi.org/10.1109/ICCD.2001.955030). URL: <http://ieeexplore.ieee.org/document/955030/> (visited on 07/16/2024) (cit. on p. 129).
- [HH15] Paul Horowitz and Winfield Hill. *The Art of Electronics*. Cambridge University Press, Mar. 2015. ISBN: 978-0-521-80926-9 (cit. on pp. 18, 27, 28, 31, 115, 151).
- [HT03] David Howe and T Tasset. "Clock Jitter Estimation based on PM Noise Measurements." In: *Proceedings of the 2003 IEEE International Frequency Control Symposium and PDA Exhibition*. 2003 Joint Mtg. IEEE Intl. Freq. Cont. Symp. and EFTF Conf, Tampa, FL, 2003-01-01 2003. URL: [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=105277](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=105277) (cit. on p. 115).
- [Hua+22] Wangwang Huang, Xuesong Mei, Yage Wang, Zhengjie Fan, Cheng Chen, and Gedong Jiang. "Two-Dimensional Phase Unwrapping by a High-Resolution Deep Learning Network." In: *Measurement* 200 (2022), p. 111566. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2022.111566>. URL: <https://www.sciencedirect.com>

- [com/science/article/pii/S0263224122007813](https://doi.org/10.1109/EMBC.2023.3252636) (cit. on p. 122).
- [Int24] Intel. *TinyCrypt*. 2024. URL: <https://github.com/intel/tinycrypt> (cit. on p. 85).
- [JG93] Howard W. Johnson and Martin Graham. *High-Speed Digital Design. A Handbook of Black Magic*. Ed. by Prentice Hall. 1993 (cit. on pp. 19, 115).
- [Kaj+23] Shugo Kaji, Daisuke Fujimoto, Masahiro Kinugawa, and Yuichi Hayashi. “Echo TEMPEST: EM Information Leakage Induced by IEMI for Electronic Devices.” In: *IEEE Transactions on Electromagnetic Compatibility* 65.3 (June 2023), pp. 655–666. ISSN: 1558-187X. DOI: [10.1109/temc.2023.3252636](https://doi.org/10.1109/temc.2023.3252636). URL: <http://dx.doi.org/10.1109/TEMC.2023.3252636> (cit. on pp. 49, 50, 146).
- [Kam24] Samy Kamkar. *Optical Espionage: Using Lasers to Hear Keystrokes Through Glass Windows*. DEF CON 2024, DEF CON Security Conference, 8-11 August 2024, Las Vegas, NV, USA. 2024 (cit. on p. 50).
- [Ken11] George Kennedy. *Electronic Communication Systems*. Ed. by McGraw-Hill Education. 5th. 2011 (cit. on p. 15).
- [Key14] Keysight. *Measuring Phase Noise with a Real Time Sampling Oscilloscope*. 2014. URL: <https://docs.keysight.com/kkbopen/measuring-phase-noise-with-a-real-time-sampling-oscilloscope-584447063.html> (cit. on p. 115).
- [KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. “Differential Power Analysis.” In: *Advances in Cryptology — CRYPTO’99*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397. ISBN: 978-3-540-48405-9 (cit. on pp. 5, 53, 55, 56, 60).
- [Koc+19] Paul Kocher et al. “Spectre Attacks: Exploiting Speculative Execution.” In: *2019 IEEE Symposium on Security and Privacy (SP)*. 2019, pp. 1–19. DOI: [10.1109/SP.2019.00002](https://doi.org/10.1109/SP.2019.00002) (cit. on p. 6).
- [Koc+20] Paul Kocher et al. “Spectre Attacks: Exploiting Speculative Execution.” In: *Commun. ACM* 63.7 (June 2020), pp. 93–101. ISSN: 0001-0782. DOI: [10.1145/3399742](https://doi.org/10.1145/3399742). URL: <https://doi.org/10.1145/3399742> (cit. on p. 6).
- [Koc96] Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems.” In: *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO’96. Berlin, Heidelberg: Springer-Verlag, 1996, pp. 104–113. ISBN: 3540615121. DOI: [10.1007/3-540-68697-5\\_9](https://doi.org/10.1007/3-540-68697-5_9) (cit. on pp. 5, 53, 56).

- [Koc+11] Paul C. Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. "Introduction to Differential Power Analysis." In: *Journal of Cryptographic Engineering* 1 (2011), pp. 5–27. URL: <https://link.springer.com/article/10.1007/s13389-011-0006-y> (cit. on pp. 56, 117).
- [Kok19] Kokke. *TinyAES: Small portable AES128/192/256 in C*. 2019. URL: <https://github.com/kokke/tiny-AES-c> (cit. on p. 132).
- [Krio07] Simon Kristiansson. "Substrate Noise Coupling in Mixed-Signal Integrated Circuits: Compact Modeling and Grounding Strategies." PhD thesis. Chalmers University of Technology, 2007. URL: <https://research.chalmers.se/en/publication/48130> (cit. on p. 29).
- [UMSo2] Un-Ku Moon, K. Mayaram, and J.T. Stonick. "Spectral Analysis of Time-Domain Phase Jitter Measurements." In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49.5 (May 2002), pp. 321–327. ISSN: 1057-7130. DOI: 10.1109/TCSII.2002.802343. URL: <http://ieeexplore.ieee.org/document/1025151/> (visited on 07/15/2024) (cit. on p. 115).
- [Kub19] Ireneusz Kubiak. "Electromagnetic Eavesdropping." In: *Recent Trends in Communication Networks*. Ed. by Pinaki Mitra. Rijeka: IntechOpen, 2019. Chap. 4. DOI: 10.5772/intechopen.86478. URL: <https://doi.org/10.5772/intechopen.86478> (cit. on p. 155).
- [Kuh03] Markus G. Kuhn. "Compromising Emanations: Eavesdropping Risks of Computer Displays." PhD thesis. University of Cambridge, 2003 (cit. on p. 52).
- [KA98] Markus G. Kuhn and Ross J. Anderson. "Soft Tempest: Hidden Data Transmission Using Electromagnetic Emanations." In: *Information Hiding*. Ed. by David Aucsmith. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 124–142. ISBN: 978-3-540-49380-8 (cit. on pp. 6, 50).
- [Lar24] Federica Laricchia. *Bluetooth Device Shipments Worldwide from 2015 to 2028*. 2024. URL: <https://www.statista.com/statistics/1220933/global-bluetooth-device-shipment-forecast/> (cit. on p. 72).
- [Lar+22] Federico Larroca, Pablo Bertrand, Felipe Carrau, and Victoria Severi. "gr-tempest: An Open-Source GNU Radio Implementation of TEMPEST." In: *2022 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)* (Dec. 2022). DOI: 10.1109/asianhost56390.2022.10022149. URL: <http://dx.doi.org/10.1109/AsianHOST56390.2022.10022149> (cit. on p. 52).

- [Lav22] Corentin Lavaud. “Reconfigurable Systems for the Interception of Compromising Sporadic Signals.” PhD thesis. Université de Rennes 1, Jan. 2022 (cit. on p. 75).
- [Lav+21] Corentin Lavaud, Robin Gerzaguët, Matthieu Gautier, Olivier Berder, Erwan Nogues, and Stéphane Molton. “Whispering Devices: A Survey on How Side-Channels Lead to Compromised Information.” In: *Journal of Hardware and Systems Security* 5 (2021), pp. 143–168. URL: <https://api.semanticscholar.org/CorpusID:233685396> (cit. on pp. 39, 40, 42–44, 49, 60).
- [LCCo8] Thanh-Ha Le, Cécile Canovas, and Jessy Clédière. “An Overview of Side Channel Analysis Attacks.” In: *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security. ASIACCS '08*. Tokyo, Japan: Association for Computing Machinery, 2008, pp. 33–43. ISBN: 9781595939791. DOI: 10.1145/1368310.1368319. URL: <https://doi.org/10.1145/1368310.1368319> (cit. on p. 99).
- [Lee99] Bang S Lee. “Understanding the Terms and Definitions of LDO Voltage Regulators.” In: (1999) (cit. on p. 130).
- [LH20] JongHyeok Lee and Dong-Guk Han. “Security Analysis on Dummy Based Side-Channel Countermeasures—Case Study: AES with Dummy and Shuffling.” In: *Applied Soft Computing* 93 (2020), p. 106352. ISSN: 1568-4946. DOI: <https://doi.org/10.1016/j.asoc.2020.106352> (cit. on p. 153).
- [Li18] Fanlong Li. *Impact of Power-Supply Noise on Phase Noise Performance of RF DACs*. Tech. rep. Texas Instruments, 2018 (cit. on p. 129).
- [LMM05] Huiyun Li, A. Theodore Markettos, and Simon Moore. “Security Evaluation Against Electromagnetic Analysis at Design Time.” In: *Proceedings of the 7th International Conference on Cryptographic Hardware and Embedded Systems. CHES'05*. Edinburgh, UK: Springer-Verlag, 2005, pp. 280–292. ISBN: 3540284745. DOI: 10.1007/11545262\_21. URL: [https://doi.org/10.1007/11545262\\_21](https://doi.org/10.1007/11545262_21) (cit. on pp. 29, 41, 43, 100, 117).
- [TP-24] TP-Link. *TL-ANT2424B*. 2024. URL: <https://www.tp-link.com/fr/home-networking/antenna/tl-ant2424b/> (cit. on p. 83).
- [LBC16] Owen Lo, William Buchanan, and Douglas Carson. “Power Analysis Attacks on the AES-128 S-box Using Differential Power Analysis (DPA) and Correlation Power Analysis (CPA).” In: *Journal of Cyber Security Technology* 1 (Sept.

- 2016), pp. 1–20. DOI: [10.1080/23742917.2016.1231523](https://doi.org/10.1080/23742917.2016.1231523) (cit. on p. 5).
- [Lyoo08] Richard Lyons. *Quadrature Signals: Complex, But Not Complicated*. <https://dspguru.com/files/QuadSignals.pdf>. 2008 (cit. on p. 17).
- [Lyoo10] Richard Lyons. *Understanding Digital Signal Processing*. 3rd. Pearson, 2010. ISBN: 978-0137027415 (cit. on p. 16).
- [Mano03] Stefan Mangard. “A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion.” In: *Information Security and Cryptology — ICISC 2002*. Ed. by Pil Joong Lee and Chae Hoon Lim. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 343–358. ISBN: 978-3-540-36552-5 (cit. on pp. 5, 34).
- [MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security)*. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN: 0387308571 (cit. on pp. 54, 55, 153).
- [MBR24] Jaakko Marin, Micael Bernhardt, and Taneli Riihonen. “Experimental Evaluation of Jamming Waveforms Against WLAN Transmission From COTS Laptop.” In: *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. 2024, pp. 1–6. DOI: [10.1109/WCNC57260.2024.10570677](https://doi.org/10.1109/WCNC57260.2024.10570677) (cit. on p. 53).
- [Mar14] Martin Marinov. “Remote Video Eavesdropping Using a Software-Defined Radio Platform.” MA thesis. University of Cambridge, 2014 (cit. on p. 52).
- [Mar11] A. T. Markettos. “Active Electromagnetic Attacks on Secure Hardware.” In: 2011 (cit. on pp. 31, 32, 47, 48).
- [MG10] Edgar Mateos and Catherine H. Gebotys. “A New Correlation Frequency Analysis of the Side Channel.” In: *Proceedings of the 5th Workshop on Embedded Systems Security. WESS '10*. Scottsdale, Arizona: Association for Computing Machinery, 2010. ISBN: 9781450300780. DOI: [10.1145/1873548.1873552](https://doi.org/10.1145/1873548.1873552). URL: <https://doi.org/10.1145/1873548.1873552> (cit. on p. 161).
- [MOW14] Luke Mather, Elisabeth Oswald, and Carolyn Whitnall. “Multi-Target DPA Attacks: Pushing DPA Beyond the Limits of a Desktop Computer.” In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by Palash Sarkar and Tetsu Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 243–261. ISBN: 978-3-662-45611-8 (cit. on pp. 59, 161).

- [Med+10] Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. “Fresh Re-Keying: Security Against Side-Channel and Fault Attacks for Low-Cost Devices.” In: *Progress in Cryptology – AFRICACRYPT 2010*. Ed. by Daniel J. Bernstein and Tanja Lange. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 279–296. ISBN: 978-3-642-12678-9 (cit. on p. 154).
- [Meu+23] Jesse De Meulemeester, Antoon Purnal, Lennert Wouters, Arthur Beckers, and Ingrid Verbauwhede. “SpectrEM: Exploiting Electromagnetic Emanations During Transient Execution.” In: *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 6293–6310. ISBN: 978-1-939133-37-3. URL: <https://www.usenix.org/conference/usenixsecurity23/presentation/de-meulemeester> (cit. on p. 51).
- [Mey12] Olivier Meynard. “Characterization and Use of the EM Radiation to Enhance Side Channel Attacks.” PhD thesis. Télécom ParisTech, Jan. 2012 (cit. on pp. 58, 59, 109, 126, 127).
- [Mic20] Microchip. *megaAVR Data Sheet: ATmega48A / PA / 88A / PA / 168A / PA / 328 / P*. 2020 (cit. on p. 131).
- [Min24] Mini-Circuits. *ZX60-272LN-S+*. 2024. URL: <https://www.minicircuits.com/pdfs/ZX60-272LN-S+.pdf> (cit. on p. 83).
- [Moh11] Habeeb Ur Rahman Mohammed. *Supply Noise Effect on Oscillator Phase Noise*. Tech. rep. Texas Instruments, 2011 (cit. on p. 129).
- [Mol11] Andreas F. Molisch. *Wireless Communications*. Second. Wiley Publishing, 2011. ISBN: 978-0-470-74186-3 (cit. on pp. 15, 25).
- [MMT23] Saleh Khalaj Monfared, Tahoura Mosavirik, and Shahin Tajik. “LeakyOhm: Secret Bits Extraction using Impedance Analysis.” In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. CCS ’23*. Copenhagen, Denmark: Association for Computing Machinery, 2023, pp. 1675–1689. ISBN: 9798400700507. DOI: [10.1145/3576915.3623092](https://doi.org/10.1145/3576915.3623092). URL: <https://doi.org/10.1145/3576915.3623092> (cit. on p. 146).
- [MSY13] Jim Monthie, Vineet Sreekumar, and Ranjit Yashwante. “Impact of Power Supply Noise on Clock Jitter in High-Speed DDR Memory Interfaces.” In: *2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*. 2013, pp. 262–266. DOI: [10.1109/VLSID.2013.198](https://doi.org/10.1109/VLSID.2013.198) (cit. on p. 130).



- [Mon+13] David P. Montminy, Rusty O. Baldwin, Michael A. Temple, and Eric D. Laspe. “Improving Cross-Device Attacks using Zero-Mean Unit-Variance Normalization.” In: *Journal of Cryptographic Engineering* 3 (2013), pp. 99–110. DOI: [10.1007/s13389-012-0038-y](https://doi.org/10.1007/s13389-012-0038-y). URL: <https://api.semanticscholar.org/CorpusID:18343838> (cit. on pp. 97, 99).
- [Mos+23] Tahoura Mosavirik, Saleh Khalaj Monfared, Maryam Saadat Safa, and Shahin Tajik. “Silicon Echoes: Non-Invasive Trojan and Tamper Detection using Frequency-Selective Impedance Analysis.” In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2023, Issue 4 (2023), pp. 238–261. DOI: [10.46586/tches.v2023.i4.238-261](https://doi.org/10.46586/tches.v2023.i4.238-261). URL: <https://tches.iacr.org/index.php/TCHES/article/view/11165> (cit. on p. 146).
- [Nas+23] Ben Nassi, Etay Iluz, Or Cohen, Ofek Vayner, Dudi Nassi, Boris Zadov, and Yuval Elovici. “Video-Based Cryptanalysis: Extracting Cryptographic Keys from Video Footage of a Device’s Power LED.” In: (2023). URL: <https://eprint.iacr.org/2023/923> (cit. on pp. 56, 119).
- [NPO8] Shahin Nazarian and Massoud Pedram. “Crosstalk-affected delay analysis in nanometer technologies.” In: *International Journal of Electronics* 95.9 (2008), pp. 903–937. DOI: [10.1080/00207210802312161](https://doi.org/10.1080/00207210802312161). eprint: <https://doi.org/10.1080/00207210802312161>. URL: <https://doi.org/10.1080/00207210802312161> (cit. on p. 146).
- [New24] NewAE. *H-Field Probe*. 2024. URL: <https://www.newae.com/products/nae-hprobe-15> (cit. on p. 131).
- [New] NewAE. *CW1173 ChipWhisperer-Lite - NewAE Hardware Product Documentation*. URL: <https://rtfm.newae.com/Capture/ChipWhisperer-Lite/> (visited on 07/15/2024) (cit. on p. 134).
- [Ngu15] Cam Nguyen. *Radio-Frequency Integrated-Circuit Engineering*. Ed. by Wiley. 2015. ISBN: 978-1-118-93648-1 (cit. on pp. 18, 20).
- [OF124] Colin O’Flynn. “Phase Modulation Side Channels: Jittery JTAG for On-Chip Voltage Measurements.” In: *IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2024.4 (Sept. 2024), pp. 382–424. DOI: [10.46586/tches.v2024.i4.382-424](https://doi.org/10.46586/tches.v2024.i4.382-424). URL: <https://tches.iacr.org/index.php/TCHES/article/view/11797> (cit. on pp. 118, 119).

- [OW21] Colin O’Flynn and Jasper van Woudenberg. *The Hardware Hacking Handbook. Breaking Embedded Security with Hardware Attacks*. Ed. by No Starch Press. 2021. ISBN: 9781593278748 (cit. on p. 54).
- [Ors+04] S.B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel. “Power-Analysis Attack on an ASIC AES Implementation.” In: *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004*. Vol. 2. 2004, 546–552 Vol.2. DOI: [10.1109/ITCC.2004.1286711](https://doi.org/10.1109/ITCC.2004.1286711) (cit. on pp. 5, 34).
- [Oul+20] Maamar Ouladj, Nadia El Mrabet, Sylvain Guilley, Philippe Guillot, and Gilles Millerieux. “On The Power of Template Attacks in Highly Multivariate Context.” In: *Journal of Cryptographic Engineering* 10 (Nov. 2020). DOI: [10.1007/s13389-020-00239-2](https://doi.org/10.1007/s13389-020-00239-2) (cit. on p. 58).
- [PDY16] Hoda Pahlevanzadeh, Jaya Dofe, and Qiaoyan Yu. “Assessing CPA Resistance of AES with Different Fault Tolerance Mechanisms.” In: *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2016, pp. 661–666. DOI: [10.1109/ASPDAC.2016.7428087](https://doi.org/10.1109/ASPDAC.2016.7428087). URL: <https://ieeexplore.ieee.org/document/7428087> (cit. on p. 60).
- [Par09] Raj S. Parihar. *Substrate Coupling Noise: Modeling and Mitigation Techniques*. Tech. rep. University of Rochester, 2009 (cit. on pp. 29, 147).
- [Pau06] Clayton R. Paul. *Introduction to Electromagnetic Compatibility*. Ed. by Wiley. 2nd Edition. 2006. ISBN: 978-0-471-75500-5 (cit. on pp. xxii, 13, 19, 26, 28, 29, 34, 50, 105, 152, 160).
- [Pi24] Raspberry Pi. *RP2040 Datasheet, A Microcontroller by Raspberry Pi*. Version 576cee3. 2024 (cit. on p. 131).
- [PZ22] Hossein Pirayesh and Huacheng Zeng. “Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey.” In: *IEEE Communications Surveys & Tutorials* 24.2 (2022), pp. 767–809. DOI: [10.1109/COMST.2022.3159185](https://doi.org/10.1109/COMST.2022.3159185) (cit. on p. 52).
- [PSG16] Romain Poussier, François-Xavier Standaert, and Vincent Grosso. “Simple Key Enumeration (and Rank Estimation) Using Histograms: An Integrated Approach.” In: *Cryptographic Hardware and Embedded Systems – CHES 2016*. Ed. by Benedikt Gierlichs and Axel Y. Poschmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 61–81. ISBN: 978-3-662-53140-2. URL: <https://link.springer>.



- com/chapter/10.1007/978-3-662-53140-2\_4 (cit. on p. 60).
- [Pro24] Zephyr Project. *Zephyr Project Documentation*. 2024. URL: <https://docs.zephyrproject.org/latest/introduction/index.html> (cit. on p. 76).
- [PR13] Emmanuel Prouff and Matthieu Rivain. "Masking Against Side-Channel Attacks: A Formal Security Proof." In: *Advances in Cryptology - EUROCRYPT 2013*. Vol. 7881. Lecture Notes in Computer Science. Springer, 2013, pp. 142–159. DOI: 10.1007/978-3-642-38348-9\_9 (cit. on p. 153).
- [Prv+17] Milos Prvulovic, Alenka Zajić, Robert L. Callan, and Christopher J. Wang. "A Method for Finding Frequency-Modulated and Amplitude-Modulated Electromagnetic Emanations in Computer Systems." In: *IEEE Transactions on Electromagnetic Compatibility*. Vol. 59. Feb. 2017, pp. 34–42. DOI: 10.1109/TEMC.2016.2603847 (cit. on pp. 44, 50, 64).
- [Put18] Sadasivan Puthusserypady. *Complex Signals*. <http://bme.elektro.dtu.dk/31610/notes/complex.signals.pdf>. 2018. URL: <http://bme.elektro.dtu.dk/31610/notes/complex.signals.pdf> (cit. on p. 17).
- [QS01] Jean-Jacques Quisquater and David Samyde. "Electromagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards." In: *Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security*. E-SMART '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 200–210. ISBN: 3540426108 (cit. on pp. 5, 41, 56, 60, 69, 118, 119, 126).
- [Ram+22] Soundarya Ramesh, Ghazali Suhariyanto Hadi, Sihun Yang, Mun Choon Chan, and Jun Han. "TickTock: Detecting Microphone Status in Laptops Leveraging Electromagnetic Leakage of Clock Signals." In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. CCS '22. New York, NY, USA: Association for Computing Machinery, Nov. 2022, pp. 2475–2489. ISBN: 978-1-4503-9450-5. DOI: 10.1145/3548606.3560698. URL: <https://doi.org/10.1145/3548606.3560698> (visited on 01/23/2023) (cit. on p. 28).
- [RD20] Mark Randolph and William Diehl. "Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman." In: *Cryptography* 4.2 (2020). ISSN: 2410-387X. DOI: 10.3390/cryptography4020015 (cit. on p. 99).
- [Raz12] Behzad Razavi. *RF Microelectronics*. Ed. by Prentice Hall. 2nd Edition. 2012. ISBN: 978-0-13-713473-1 (cit. on pp. 18, 30–33).

- [Res] Ettus Research. *USRP B210 SDR Kit - Dual Channel Transceiver (70 MHz - 6GHz)*. URL: <https://www.ettus.com/all-products/ub210-kit/> (cit. on pp. 83, 101).
- [RS15] Payl Reuvers and Marc Simons. *The Great Seal Bug: The Thing*. 2015. URL: <https://cryptomuseum.com/covert/bugs/thing/index.htm> (cit. on p. 49).
- [Rob19] Pieter Robyns. "Explicit and Implicit Information Leakage in Wireless Communication." PhD thesis. Hasselt University, Belgium, 2019. URL: <https://hdl.handle.net/1942/30090> (cit. on p. 91).
- [RP03] John Rogers and Calvin Plett. *Radio Frequency Integrated Circuit Design*. Ed. by Artech House. 2003. ISBN: 1-58053-502-x (cit. on p. 30).
- [Ros82] Howard E. Rosenblum. *NACSIM 5000: Tempest Fundamentals*. Tech. rep. National Security Agency (NSA), 1982. URL: <https://cryptome.org/jya/nacsim-5000/nacsim-5000.htm> (cit. on pp. 5, 34, 39, 41-43, 45-47, 117).
- [Ros20] Ross Anderson. *Security Engineering*. 3rd Edition. 2020. URL: <https://www.cl.cam.ac.uk/~rja14/book.html> (cit. on p. 54).
- [Rug15] Jan Sönke Ruge. *Side-channel Attacks on RSA Using a Software-Defined Radio*. 2015. URL: <https://bolek42.github.io/sca/introduction.html#side-sca> (cit. on p. 5).
- [Rup20] David Rupperecht. "Enhancing the Security of 4G and 5G Mobile Networks on Protocol Layer Two." PhD thesis. Ruhr-Universität at Bochum, May 2020 (cit. on p. 154).
- [SN20] Annamaria Sarbu and Dumitru Neagoie. "Wi-Fi Jamming Using Software Defined Radio." In: *International Conference Knowledge-Based Organization* 26 (June 2020), pp. 162-166. DOI: 10.2478/kbo-2020-0132 (cit. on p. 53).
- [Sch+10] Othmar Schimmel, Paul Duplys, Jan Hayek, and Wolfgang Rosenstiel. "Correlation Power Analysis in Frequency Domain." In: *Proceedings of the 1st International Workshop on Constructive Side-Channel Analysis and Secure Design*. COSADE 2010. 2010, pp. 1-3. URL: <https://api.semanticscholar.org/CorpusID:63165412> (cit. on p. 161).
- [Scho2] Ron Schmitt. *Electromagnetics Explained. A Handbook for Wireless/RF, EMC, And High-Speed Electronics*. Ed. by Newnes (Elsevier Science). 2002. ISBN: 0-7506-7403-2 (cit. on pp. 20, 24-27).

- [Sch96] Bruce Schneier. *Applied cryptography - Protocols, Algorithms, and Source Code in C, 2nd Edition*. Wiley, 1996. ISBN: 978-0-471-11709-4. URL: <https://www.worldcat.org/oclc/32311687> (cit. on p. 4).
- [Sch+23] Kai Schoos, Sergej Meschkov, Mehdi B. Tahoori, and Dennis R. E. Gnad. "JitSCA: Jitter-based Side-Channel Analysis in Picoscale Resolution." In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2023.3 (June 2023), pp. 294–320. DOI: [10.46586/tches.v2023.i3.294-320](https://doi.org/10.46586/tches.v2023.i3.294-320). URL: <https://tches.iacr.org/index.php/TCHES/article/view/10965> (cit. on pp. 117–119, 130).
- [SDR24] SDRplay. *RSPdx: Multi-Antenna Port 14-bit SDR*. Version 1.4. 2024. URL: <https://www.sdrplay.com/rspdx/> (cit. on p. 131).
- [Sem13] Nordic Semiconductor. *nRF51422 Product Specification*. Version 2.1. 2013 (cit. on p. 131).
- [Sem21] Nordic Semiconductor. *nRF52832 Product Specification*. Version 1.8. 2021 (cit. on pp. 101–103, 131).
- [Sem24] Nordic Semiconductor. *nRF52 Series: SoftDevices*. 2024. URL: [https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct\\_nrf52%2Fstruct%2Fnrf52\\_softdevices.html](https://infocenter.nordicsemi.com/index.jsp?topic=%2Fstruct_nrf52%2Fstruct%2Fnrf52_softdevices.html) (cit. on p. 75).
- [Sha79] Adi Shamir. "How to Share a Secret." In: *Commun. ACM* 22.11 (Nov. 1979), pp. 612–613. ISSN: 0001-0782. DOI: [10.1145/359168.359176](https://doi.org/10.1145/359168.359176). URL: <https://doi.org/10.1145/359168.359176> (cit. on p. 153).
- [She+21] Cheng Shen, Tian Liu, Jun Huang, and Rui Tan. *When LoRa Meets EMR: Electromagnetic Covert Channels Can Be Super Resilient*. 2021. URL: <https://www.computer.org/csdl/proceedings-article/sp/2021/893400a529/1oak96ToN0c> (visited on 11/24/2020) (cit. on pp. 44, 51).
- [SR92] E. Sicard and A. Rubio. "Analysis of Crosstalk Interference in CMOS Integrated Circuits." In: *IEEE Transactions on Electromagnetic Compatibility* 34.2 (1992), pp. 124–129. DOI: [10.1109/15.135625](https://doi.org/10.1109/15.135625) (cit. on pp. 27, 146).
- [Sig20] Siglent. *SDS 2000X-Plus Datasheet*. [https://siglentna.com/wp-content/uploads/dlm/uploads/2021/03/SDS2000X-Plus-Datasheet\\_DS0102XP\\_E01B.pdf](https://siglentna.com/wp-content/uploads/dlm/uploads/2021/03/SDS2000X-Plus-Datasheet_DS0102XP_E01B.pdf). 2020. URL: [https://siglentna.com/wp-content/uploads/dlm/uploads/2021/03/SDS2000X-Plus-Datasheet\\_DS0102XP\\_E01B.pdf](https://siglentna.com/wp-content/uploads/dlm/uploads/2021/03/SDS2000X-Plus-Datasheet_DS0102XP_E01B.pdf) (cit. on p. 133).
- [Smi99] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. 2nd. California Technical Publishing, 1999. ISBN: 0966017676 (cit. on p. 16).

- [SBY00] C.M. Spooner, W.A. Brown, and G.K. Yeung. “Automatic radio-frequency environment analysis.” In: *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No.00CH37154)*. Vol. 2. 2000, 1181–1186 vol.2. DOI: [10.1109/ACSSC.2000.910700](https://doi.org/10.1109/ACSSC.2000.910700) (cit. on p. 64).
- [SMC20] Albert Spruyt, Alyssa Milburn, and Lukasz Chmielewski. “Fault Injection as an Oscilloscope: Fault Correlation Analysis.” In: *IACR Transactions on Cryptographic Hardware and Embedded Systems*. TCHES’21 2021.1 (Dec. 2020), pp. 192–216. DOI: [10.46586/tches.v2021.i1.192-216](https://doi.org/10.46586/tches.v2021.i1.192-216). URL: <https://tches.iacr.org/index.php/TCHES/article/view/8732> (cit. on pp. 54, 119).
- [Sta10] François-Xavier Standaert. “Introduction to Side-Channel Attacks.” In: *Secure Integrated Circuits and Systems*. Ed. by Ingrid M.R. Verbauwhede. Boston, MA: Springer US, 2010, pp. 27–42. ISBN: 978-0-387-71829-3. DOI: [10.1007/978-0-387-71829-3\\_2](https://doi.org/10.1007/978-0-387-71829-3_2). URL: [https://doi.org/10.1007/978-0-387-71829-3\\_2](https://doi.org/10.1007/978-0-387-71829-3_2) (cit. on p. 5).
- [SN01] National Institute of Standards and Technology (NIST). *Advanced Encryption Standard (AES)*. Federal Information Processing Standard (FIPS) 197. Nov. 2001. DOI: [10.6028/NIST.FIPS.197](https://csrc.nist.gov/publications/detail/fips/197/final). URL: <https://csrc.nist.gov/publications/detail/fips/197/final> (visited on 02/28/2023) (cit. on p. 34).
- [SF] Steven Chen and Frank Kearney. *Passive Intermodulation (PIM) Effects in Base Stations: Understanding the Challenges and Solutions*. URL: <https://www.analog.com/en/analog-dialogue/articles/passive-intermodulation-effects-in-base-stations-understanding-the-challenges-and-solutions.html> (cit. on p. 32).
- [STM18] STMicroelectronics. *Datasheet - STM32F103xC, STM32F103xD, STM32F103xE*. Version DS5792 Rev 13. 2018 (cit. on p. 132).
- [STM21] STMicroelectronics. *Datasheet - STM32L151xE STM32L152xE*. Version DS10002 Rev 10. 2021 (cit. on p. 131).
- [STM24] STMicroelectronics. *STM32 Nucleo-64 Development Board with STM32F103RB MCU*. Version DB2196 - Rev 19. 2024. URL: <https://www.st.com/en/evaluation-tools/nucleo-f103rb.html> (cit. on p. 132).
- [Su+17] Yang Su, Daniel Genkin, Damith Ranasinghe, and Yuval Yarom. “USB Snooping Made Easy: Crosstalk Leakage Attacks on USB Hubs.” In: *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1145–1161. ISBN: 978-1-931971-40-9.

- URL: <https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/su> (cit. on p. 40).
- [Sul+90] Donald B. Sullivan, David W. Allan, David A. Howe, and Fred L. Walls. *Characterization of Clocks and Oscillators*. Tech. rep. Mar. 1990, p. 22539 (cit. on p. 115).
- [TW11] Andrew S. Tanenbaum and David J. Watherall. *Computer Networks*. Ed. by Prentice Hall. 5th. 2011. ISBN: 0132553171 (cit. on p. 4).
- [Tek24a] TekBox. *TBPS01 probe*. 2024. URL: <https://www.tekbox.com/product/tekbox-tbps01-emc-near-field-probes/> (cit. on pp. 83, 131).
- [Tek24b] TekBox. *TBWA2: Wideband RF Amplifiers*. Version 1.3. 2024 (cit. on p. 132).
- [Tek05] Tektronix. *Jitter Analysis: A Brief Guide to Jitter*. Tech. rep. Tektronix, 2005 (cit. on p. 115).
- [Thio1] Erik Thiele. *Tempest For Eliza*. 2001. URL: <http://www.erikyyy.de/tempest/> (cit. on p. 51).
- [TE98] J. Thornton and D.J. Edwards. "Modulating Retro-Reflector as a Passive Radar Transponder." In: *Electron. Lett.* 34.19 (1998), p. 1880. ISSN: 00135194. DOI: 10.1049/el:19981326. URL: [https://digital-library.theiet.org/content/journals/10.1049/el\\_19981326](https://digital-library.theiet.org/content/journals/10.1049/el_19981326) (visited on 04/27/2022) (cit. on p. 49).
- [TCA14] Kevin Townsend, Carles Cuf, and Akiba. *Getting Started with Bluetooth Low Energy. Tools and Techniques for Low-Power Networking*. Ed. by O'Reilly. 2014. ISBN: 978-1-4919-4951-1 (cit. on p. 72).
- [Toy19] Ichihiko Toyoda. "Polarization Modulation." In: *Modulation in Electronics and Telecommunications*. 2019 (cit. on p. 109).
- [Van85] Wim Van Eck. "Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?" In: *North-Holland Computers & Security* (1985), pp. 269–286. DOI: 10.1016/0167-4048(85)90046-X (cit. on pp. 5, 51).
- [VGS13] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. "Security Evaluations Beyond Computing Power." In: *Advances in Cryptology – EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 126–141. ISBN: 978-3-642-38348-9. URL: [https://link.springer.com/chapter/10.1007/978-3-642-38348-9\\_8](https://link.springer.com/chapter/10.1007/978-3-642-38348-9_8) (cit. on p. 60).

- [Vor03] Ashish C Vora. *Substrate Coupling in RF Analog/Mixed Signal IC Design: A Review*. Tech. rep. Rochester Institute of Technology, 2003 (cit. on p. 29).
- [VP09] Martin Vuagnoux and Sylvain Pasini. “Compromising Electromagnetic Emanations of Wired and Wireless Keyboards.” In: *Proceedings of the 18th Conference on USENIX Security Symposium*. SSYM’09. Montreal, Canada: USENIX Association, 2009, pp. 1–16 (cit. on pp. 6, 41, 116, 119, 161).
- [Wag01] David Wagner. *CryptoMe NONSTOP And HIJACK Possible Meanings*. 2001. URL: <https://cryptome.org/nonstop-hijack.htm> (cit. on p. 46).
- [Wal14] Jearl Walker. *Fundamentals of Physics*. Ed. by Wiley. 10th Extended Edition. 2014. ISBN: 978-1-118-23072-5 (cit. on pp. 20, 22, 23, 25).
- [Wan+16] Christopher Wang, Robert Callan, Alenka Zajic, and Milos Prvulovic. “An Algorithm for Finding Carriers of Amplitude-Modulated Electromagnetic Emanations in Computer Systems.” In: *2016 10th European Conference on Antennas and Propagation (EuCAP)*. Apr. 2016, pp. 1–5. DOI: 10.1109/EuCAP.2016.7481633 (cit. on pp. 44, 50, 64).
- [Wan+21] Meizhi Wang, Vishnuvardhan V. Iyer, Shanshan Xie, Ge Li, Sanu K. Mathew, Raghavan Kumar, Michael Orshansky, Ali E. Yilmaz, and Jaydeep P. Kulkarni. “Physical Design Strategies for Mitigating Fine-Grained Electromagnetic Side-Channel Attacks.” In: *2021 IEEE Custom Integrated Circuits Conference (CICC)*. 2021, pp. 1–2. DOI: 10.1109/CICC51472.2021.9431438 (cit. on p. 105).
- [WWD20] Ruize Wang, Huanyu Wang, and Elena Dubrova. “Far Field EM Side-Channel Attack on AES Using Deep Learning.” In: *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security* (Nov. 2020). DOI: 10.1145/3411504.3421214. URL: <http://dx.doi.org/10.1145/3411504.3421214> (cit. on pp. 62, 116, 119).
- [War67] Willis H. Ware. “Security and Privacy in Computer Systems.” In: *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*. AFIPS ’67 (Spring). Atlantic City, New Jersey: Association for Computing Machinery, 1967, pp. 279–282. ISBN: 9781450378956. DOI: 10.1145/1465482.1465523. URL: <https://www.rand.org/pubs/papers/P3544.html> (cit. on p. 37).
- [Wei+15] Teng Wei, Shu Wang, Anfu Zhou, and Xinyu Zhang. “Acoustic Eavesdropping through Wireless Vibrometry.” In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. MobiCom ’15. Paris, France:



- Association for Computing Machinery, 2015, pp. 130–141. ISBN: 9781450336192. DOI: [10.1145/2789168.2790119](https://doi.org/10.1145/2789168.2790119). URL: <https://doi.org/10.1145/2789168.2790119> (cit. on p. 50).
- [Wer+18] Frank Werner, Derrick Albert Chu, Antonije R. Djordjević, Dragan I. Olćan, Milos Prvulovic, and Alenka Zajić. “A Method for Efficient Localization of Magnetic Field Sources Excited by Execution of Instructions in a Processor.” In: *IEEE Transactions on Electromagnetic Compatibility* 60.3 (2018), pp. 613–622. DOI: [10.1109/TEM.2017.2742501](https://doi.org/10.1109/TEM.2017.2742501) (cit. on p. 64).
- [Wit23] Marc Witteman. *Security Highlight: You May Be Leaking Secrets if You Don't Keep Your Pace*. Riscure. 2023. URL: <https://www.riscure.com/security-highlight-clock-jitter/> (cit. on p. 118).
- [Wu+24] J. Wu, R. Wu, D. Xu, D. Tian, and A. Bianchi. “SoK: The Long Journey of Exploiting and Defending the Legacy of King Harald Bluetooth.” In: *2024 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2024, pp. 23–23. DOI: [10.1109/SP54263.2024.00023](https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00023). URL: <https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00023> (cit. on p. 80).
- [Xav99] Antonio Rubio Xavier Aragonès José Luis Gonzalez. *Analysis and Solutions for Switching Noise Coupling in Mixed-Signal ICs*. Ed. by Springer. 1999. ISBN: 978-1-4419-5085-7. DOI: [10.1007/978-1-4757-3013-5](https://link.springer.com/book/10.1007/978-1-4757-3013-5). URL: <https://link.springer.com/book/10.1007/978-1-4757-3013-5> (cit. on p. 29).
- [Yan+17] Wei Yang, Yongbin Zhou, Yuchen Cao, Hailong Zhang, Qian Zhang, and Huan Wang. “Multi-Channel Fusion Attacks.” In: *Trans. Info. For. Sec.* 12.8 (Aug. 2017), pp. 1757–1771. ISSN: 1556-6013. DOI: [10.1109/TIFS.2017.2672521](https://doi.org/10.1109/TIFS.2017.2672521). URL: <https://doi.org/10.1109/TIFS.2017.2672521> (cit. on p. 59).
- [YK18] Weize Yu and Selçuk Köse. “Exploiting Voltage Regulators to Enhance Various Power Attack Countermeasures.” In: *IEEE Transactions on Emerging Topics in Computing* 6.2 (2018), pp. 244–257. DOI: [10.1109/TETC.2016.2620382](https://doi.org/10.1109/TETC.2016.2620382) (cit. on p. 147).
- [ZP23] Alenka Zajic and Milos Prvulovic. *Understanding Analog Side Channels Using Cryptography Algorithms*. Ed. by Springer. 2023. ISBN: 978-3-031-38578-0 (cit. on p. 51).

- [ZP14] Alenka Zajić and Milos Prvulovic. “Experimental Demonstration of Electromagnetic Information Leakage From Modern Processor-Memory Systems.” In: *IEEE Transactions on Electromagnetic Compatibility*. Vol. 56. Aug. 2014, pp. 885–893. DOI: [10.1109/TEMC.2014.2300139](https://doi.org/10.1109/TEMC.2014.2300139) (cit. on pp. 44, 50, 51, 64).
- [Zed22] Axel Zedigh. “Improving Deep Learning Assisted Far-Field Electromagnetic Side-Channel Attacks on AES.” MA thesis. KTH Royal Institute of Technology, 2022 (cit. on p. 5).
- [ZF05] YongBin Zhou and DengGuo Feng. *Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing*. 2005. URL: <http://eprint.iacr.org/2005/388> (cit. on p. 55).



# INDEX

- Frequency hopping, 86, 90
- fault correlation analysis, 118
- GFSK, 71
- CE, 87, 90
- CMOS, 100
- CO, 86, 91
- EMC, 100
- LNA, 83
- LTK, 70, 79, 80
- MD, 80, 87, 90
- MSoC, 7, 61
- PGE, 94
- SKD, 80, 93
- BLE, 69, 71
- SDR, 3, 83
- illumination, 61
- AddRoundKey, 93
- BD\_ADDR, 80
- CONNECT\_IND, 87, 90
- CONNECT, 80
- EDIV, 79, 80
- HCLK64M, 100
- LL\_ENC\_REQ, 79, 80, 87, 89
- LL\_START\_ENC\_REQ, 89
- PCLK32M, 100
- RAND, 79, 80
- SubBytes, 93
  
- Acoustic attack, 5
- AES, 5, 70, 93
- Air-gapped exfiltration, 6
- ASCAD, 161
  
- BlueScream, 9, 10
  
- Cache, 103
- Cache side channels, 6
- CIA, 5
- Clock, 100
- Connection Event, 74
- Covert channel, 1
  
- Covert channels, 6
- Cross-correlation, 91
- Crosstalk, 7, 37
  
- DES, 5
- Diffie-Hellman, 4
  
- Eddystone, 70, 76, 90
- Electromagnetic attack, 5
  
- Firmware, 75, 85
- Flash controller, 103
- Frequency diversity, 104
- Frequency hopping, 75
  
- GPIO, 86
  
- Harmonic, 100
- HCI, 75
- Hop Interval, 90
  
- Inter-modulation, 100
- Interleaved Procedure, 89, 90
  
- Kerckhoff's principle, 4
- Key Rank, 94
  
- LE Legacy Pairing, 72, 79
- LE Secure Connection, 72
- LE Secure Connections, 70, 79
- Leakage characterization, 100, 103
- Leakage detection, 91
- Long-Term Key, 72
- Low-level traffic injection, 80, 87, 90, 103
  
- Maxwell equation, 2
- Micro-benchmark, 103
- Modulation, 109
- Mono-channel attack, 125
- Multi-channel attack, 127
  
- NimBLE, 75, 78, 85

- Noise-SDR, 6
- NSA, 5
- Out-of-Band Signal Injection, 6
- Pairing, 72
- Phase modulation, 104
- PhaseSCA, 9, 10
- Polarization, 109
- Power attack, 5
- Privacy, 37
- Private, 37
- Probe, 83
- Profile, 96, 99
- Profiled Correlation attack, 94
- Radiation, 37
- Reproducibility, 161
- RSA, 5
- Screaming Channels, 6, 69, 70
- Session key, 73
- Shielding, 105
- Side-channel attack, 4
- SKEBLE, 77
- Sniff, 79
- Soft-TEMPEST, 6
- SoftDevice, 75
- Space diversity, 104
- Spectre, 6
- Substrate coupling, 7, 61, 100
- TekBox, 83
- TEMPEST, 5
- Template attack, 80, 82, 94, 96, 99
- Timing attack, 5
- TinyCrypt, 78
- USRP, 83
- Van Eck Phreaking, 5
- WazaBee, 6
- Zephyr, 75

## COLOPHON

This document was typeset in L<sup>A</sup>T<sub>E</sub>X using the `classicthesis` style developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's book on typography "*The Elements of Typographic Style*".

*Final Version* as of March 2, 2025 (1.0).

