Understanding Screaming Channels: From a Detailed Analysis to Improved Attacks

Giovanni Camurati, Aurélien Francillon, François-Xavier Standaert

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2020, No. 3, pp. 358–401. DOI:10.13154/tches.v2020.i3.358-401

Understanding Screaming Channels: From a Detailed Analysis to Improved Attacks

Giovanni Camurati[1], Aurélien Francillon[1] and François-Xavier Standaert[2]

[1] EURECOM, Sophia-Antipolis, France, name.surname@eurecom.fr
[2] Université catholique de Louvain, Louvain-la-Neuve, Belgium, fstandae@uclouvain.be

Abstract. Recently, some wireless devices have been found vulnerable to a novel class of side-channel attacks, called Screaming Channels. These leaks might appear if the sensitive leaks from the processor are unintentionally broadcast by a radio transmitter placed on the same chip. Previous work focuses on identifying the root causes, and on mounting an attack at a distance considerably larger than the one achievable with conventional electromagnetic side channels, which was demonstrated in the low-noise environment of an anechoic chamber. However, a detailed understanding of the leak, attacks that take full advantage of the novel vector, and security evaluations in more practical scenarios are still missing. In this paper, we conduct a thorough experimental analysis of the peculiar properties of Screaming Channels. For example, we learn about the coexistence of intended and unintended data, the role of distance and other parameters on the strength of the leak, the distortion of the leak model, and the portability of the profiles. With such insights, we build better attacks. We profile a device connected via cable with $10000 \cdot 500$ traces. Then, 5 months later, we attack a different instance at 15 m in an office environment. We recover the *AES-128* key with $5000 \cdot 1000$ traces and key enumeration up to 2^{23}. Leveraging spatial diversity, we mount some attacks in the presence of obstacles. As a first example of application to a real system, we show a proof-of-concept attack against the authentication method of Google Eddystone beacons. On the one side, this work lowers the bar for more realistic attacks, highlighting the importance of the novel attack vector. On the other side, it provides a broader security evaluation of the leaks, helping the defender and radio designers to evaluate risk, and the need of countermeasures.

Keywords: Side-Channel Attacks · Screaming Channels · Electromagnetic Leakage

Introduction

Modern information systems are more and more connected, often by means of wireless protocols, resulting in an increased attack surface. In particular, given the shared nature of the transmission medium, the radio link is at risk of becoming a propagation vector for sensitive information leaks. Encryption at the link layer and above constitutes a first line of defense for the secrecy of the transmitted data. However, the channel opened at the physical layer might as well transmit other side signals that carry sensitive information.

Recently, we have discovered a novel attack vector, called Screaming Channels [CPM+18], introduced by radio transmitters on mixed-signal chips. In this type of devices, very popular to add wireless (e.g., Bluetooth, WiFi) capabilities to connected objects, a processor and one or more radios lay on the same silicon chip. Since many coupling effects can arise between digital logic (processor) and analog/radio-frequency logic (radio), noise can easily flow between the two. Therefore, it can happen that sensitive data treated by the processor accidentally modulates the carrier emitted intentionally by the radio. As a consequence, the leak is broadcast at a potentially large distance. An attacker could, for example, pick this leak to mount a side-channel attack against a cryptographic algorithm used to secure the wireless communication.

Received: 2020-01-15 Accepted: 2019-03-15 Published: 2020-06-19

In [CPM+18], we focused on providing some initial examples of such remote attacks at 10 m (in an anechoic chamber), a considerable distance compared to previous work on electromagnetic side channels, and we started to uncover the physical root cause of the problem in mixed-signal chips. However, many questions are still open about the properties of the novel physical vector compared to conventional electromagnetic leaks, about the best way to exploit it in more challenging environments, for example including real-world protocols and systems.

Methodology. In this paper, we explore the leak introduced by Screaming Channels taking three main points of view.

Following the approach of natural sciences, we conduct an extensive empirical study of the novel leak, that we compare to conventional side channels. The analysis of carefully designed experiments helps us formulating and confirming our hypothesis. To this purpose, we borrow concepts and techniques from radio and side channels theory.

Taking the point of view of an attacker, we look for the best ways to exploit the leak. On the one hand, we optimize the reception and preprocessing of the radio leaks. On the other hand, we find the attack techniques that best fit the properties of the leak. We conduct several attacks in realistic and challenging environments, to confirm our choices and to assess with concrete examples the potential of the novel side channel.

Finally, from the perspective of a system security analyst, we conduct a preliminary assessment of the risk introduced by the leak on a real system, showing a proof-of-concept attack on the authentication method of Google Eddystone beacons.

Contributions. Our novel contributions are:

- A thorough investigation of the leak that we discovered in [CPM+18], which answers to the following research questions:
  - What is the relation between unintended screaming-channel leaks and the intended radio transmission? How to model the channel and optimize reception?
  - What is the difference between screaming and conventional side channels in terms of strength the leak and in terms of shape (i.e., leak model)? What is the impact of distance, frequency, and setup on these points? Can profiles be reused across distances, frequencies, and setups, and against different instances of the target device?
  - What are the best options for trace collection and preprocessing? Are time, spatial, and frequency diversity important?
  - Are the attacks still possible in challenging and realistic environments, against optimized code, and against hardware cryptographic blocks? Can the attacks be applied to real protocols and systems?
- The observation that the hardware *AES-128* block is more difficult to attack. The leak from the internal operations is not visible, preventing Differential Power Analysis (DPA). The memory transfer of the key leaks: it can be attacked with Simple Power Analysis (SPA), though key recovery was not possible with the current setup.
- A significant step forward towards realistic key recovery attacks against software *AES-128*. In particular, we show that it is possible to:
  - Profile a device at short distance in convenient conditions (connection via cable), and attack another instance at 10 m and 15 m via radio in an office environment.
  - Detect the leak at 34 m and extract traces at 60 m in an office environment.
  - Attack a device at 0.55 m in a challenging home environment with obstacles.
  - Attack the authentication method used in Google Eddystone beacons.

Related Work. Electromagnetic side channels are particularly effective when they result from the modulation of a carrier signal that radiates from the device [AARR02]. As discussed by the authors, the harmonics of the clock are usually one of the strongest signals emitted by the device, but also other intended communication signals might act as carrier for a leak. This is also mentioned in a TEMPEST document [NSA82], but the details are redacted. Radio transmitters, which intentionally emit strong intended carriers, are therefore a potential vector for information leaks. In Screaming Channels [CPM+18], the unintended side-channel leaks generated by the digital part of a mixed-signal chip modulate the carrier generated in the analog part. This is relevant for modern connected devices, given the popularity of the mixed-signal architecture for many wireless protocols. A radio front-end could also be actively exploited by an attacker to exfiltrate data from a device. In Second Order Soft-TEMPEST attacks [CEK18, ECK19], malicious software running on a device modulates an intended radio carrier to transmit sensitive data. This could be achieved in several ways: malicious drivers, hardware trojans, or unintended electromagnetic compatibility effects (cross talk, coupling, etc.), possibly amplified with hardware modifications before or after production. The authors study possible polyglot modulations and build a testbed for experimentation.

Differently from the previously mentioned attacks, conventional electromagnetic side channels are due to weak unintended emanations, and they can be exploited only from short distances. Previous work [MGDS10] has studied the impact of distance up to 0.5 m on correlation attacks against an *AES-128* implementation on FPGA. As distance increases, more amplification is required to compensate the quadratic decrease of the signal. Very close to the device, the noise from other components is stronger. Interestingly, the more the antenna is distant from the target, the more the leak model is distorted. After 0.05 m, the Hamming model becomes less and less relevant. To solve this problem, the authors estimate the model directly, reducing by a factor of ten the number of traces needed for full recovery at 0.5 m. We can infer that for conventional side channels profiles cannot be easily reused to attack at different distances. In this paper we will study the effect of distance up to 10 m on Screaming Channels, and we will discover that in this case the distortion is constant. Leaks in the far-field have also been observed on the harmonic of the clock for 2048-bit exponentiation in [AARR02], up to 12 m. The authors discuss the degradation of the signal with distance, and the feasibility of attacks at 4.5 m. In this paper, we will see that *AES-128* traces can be extracted even at 60 m.

In a realistic scenario, the attacker cannot access the target device, which is located at a considerable distance, in a possibly complex environment. The attacker would prefer to profile a different device in a more favorable setup (e.g., smaller distance, less noisy environment, simpler receiver). Comparison among profiles and the effects of device (and setup/conditions) variations on profiled attacks have been investigated by a large corpus of literature. In this paper we mainly use profiled correlation attacks, therefore we focus on trace normalization rather than on machine learning based approaches. Literature shows that, while inter-device variations and other differences have a huge impact on the performance of vanilla template attacks, using simple normalization techniques (e.g., offset removal and variance normalization) can significantly reduce the problem [MBTL13, EG12, CK14, HOTM14, CK14]. Templates built from multiple devices are more robust, but they are less effective for a single device [RSV+11, HOTM14, CK14]. It might be better to build a profile on-the-fly from a portion of the attack traces [RSV+11]. Using per-traces normalization, we achieve good portability of our profiles for different distances, setups, and instances of the target.

Outline. We first provide some background and an overview of our experimental setup in Section 1. Then we conduct a thorough analysis of the channel in Section 2. In Section 3 we first draw some lessons from the analysis of the channel, and then we attack challenging targets. In Section 4 we show a proof-of-concept attack against the authentication method of Google Eddystone beacons. Finally, we discuss our results and their replicability in Section 5 and Section 6.

1 Background

In the following we briefly recall the main notation and concepts that we will use throughout the paper.

1.1 Notations

We call a random variable X, its realization x, average $\bar{x}$, and estimate $\hat{x}$. We denote with k and p the generic sub-key and sub-plaintext in a divide and conquer attack. Given an intermediate value $y = y(p,k)$ called leak variable, $l(y)$ is the (unknown) leakage function, whereas $model(y)$ is a model of this leak. Usually, the leak is a time trace $l_y(t)$, but only a few of the points, called Points Of Interest (POIs), are informative. In Screaming Channels, a trace $l_{\bar{m},y_{p,k}}(t)$ is the average of m traces measured with the same plaintext-key pair. When referring to n traces $l_{\bar{m}}$, we will say $n \cdot m$ traces. $\mathcal{L}_p$ is a set of profiling traces l^i collected with known random plaintexts and known random keys, whereas $\mathcal{L}_a$ is an attack set collected with known random plaintexts and a fixed unknown key. Alternatively, a set $\mathcal{L}$ could be split on the fly into two disjoint sets $\mathcal{L}_p$ and $\mathcal{L}_a$. A set $\mathcal{L}$ can be partitioned into 2^n classes $\mathcal{L}_y$ according to the values of an n-bit leak variable y. Given a variable x, $\tilde{x}$ is the true value, whereas x_g is a guess or hypothesis.

1.2 Screaming Channels

In [CPM+18] we presented Screaming Channels, a side channel that may appear if the operation of a radio transmitter is impacted by a nearby digital block, letting the attacker retrieve sensitive information by simply observing the radio signal at a large distance from the target. We focused on mixed-signal chips, pointing out that the root cause of the problem is the coupling between digital blocks and the radio. Additionally, we developed proof-of-concept attacks on the `tinyAES` implementation (running at 10 m in an anechoic chamber), and on the `mbedTLS` implementation (running at 1 m in a normal environment). The main target was a Nordic Semiconductor nRF52832 (which is easy to program and offers simple access to low-level radio features), but we also observed signs of leaks on another mixed-signal chip (Qualcomm Atheros AR9271).

Note on terminology. The terms *conventional* and *screaming-channel* distinguish between classic attacks on unintended EM emissions, and attacks that exploit the modulation of an intended radio signal. We refer to the paper *Screaming Channels* [CPM+18] in upper case, whereas we use lowercase for the general term *screaming channel(s)*. An *intended radio carrier* (e.g., at 2.4 GHz) is both intentionally modulated to transmit data packets (e.g., Bluetooth) and unintentionally modulated by a *leak trace* $l_{y_{p,k}}(t)$ (e.g., corresponding to an *AES-128* encryption). The actual *leak signal* that brings sensitive information is the difference among leak traces that correspond to different values of the leak variable y, as explained in Section 1.3.

1.3 Leakage Detection and Points Of Interest

If we use a valid leak variable y to partition a sufficiently large trace set $\mathcal{L}$, the classes $\mathcal{L}_y$ will show a significant statistical difference at the points that leak (POIs). Examples of tools used for leakage/POIs detection (or as distinguishers) are the t-test [GGJR+11], the χ^2-test [MRSS18], the ρ-test [DS16], the sum of differences ([CRR02]), and the Mutual Information (MI) [GBTP08, MOBW13]. In this paper we focus on:

- **fixed vs. fixed t-test** [DS16]. Given two random variables X and Y the Welch's t-test [Wel47] checks whether they exhibit a significant difference or not. So it can be naturally used to measure if a difference in the key produces a difference in the leak. It is defined as $(\bar{X} - \bar{Y})/\sqrt{\frac{var(X)}{N_x} + \frac{var(Y)}{N_y}}$, where N is the sample size. Many strategies

have been proposed and used in literature (e.g., [GGJR+11, BCD+13, MOBW13]) for the choice of the two sets to analyze. One of the most common choices is the non-specific random vs. fixed test, which compares a set of traces collected with random plaintext and key with a set for which the key is fixed. To improve detection, the authors of [DS16] propose a fixed vs. fixed test that compares two sets for which the difference is supposed to be maximal (e.g., by choosing two very different values of the Hamming Weight of the leak variable).

- **k-fold ρ-test** [DS16]. The ρ-test aims at first learning the (possibly nonlinear) model $model(y)$ of the leak function using a profiling set, and then measuring the (linear) correlation of the learned model with a set of test traces. More precisely, profiling traces are classified according to the possible values of the leak variable, and the average of each class is used as model of the leak for that value of the leak variable: $\hat{model}_Y(y) = \bar{L}_{p,y}$. Then, the Pearson's correlation coefficient is used to measure how well this model correlates with the test traces: $\hat{r} = \hat{\rho}(L(y), \hat{model}_Y(y))$. To reduce the bias, the results obtained from k different partitions of the starting set into profiling and attack sets are averaged together. With N traces, $\hat{r}_z = \frac{1}{2\sqrt{N-3}} ln(\frac{1+\hat{r}}{1-\hat{r}})$ approximately follows a normal distribution $\mathcal{N}(0,1)$ and can be used to to measure the confidence that data-dependent information is leaking.

1.4 Profiled Attacks

An attacker with access to (another instance of) the target device can profile it (i.e., estimate the distribution of the leak in a controlled setting). This knowledge is valuable to understand the physical behavior of the chip, and to compare different setups. After this offline phase, the attacker can collect traces with unknown key on the actual target, and leverage the profile to find the most likely value of the key. In the following we briefly recall profiled correlation and template attacks.

- **Profiled Correlation Attacks** [DS16]. Using the profiling traces, the attacker estimates the model $\hat{model}(y) = \bar{L}_{t_{poi}}(y)$ for each value of y. Then, the attacker measures the correlation between the attack traces and the values indicated by the precomputed profile for a given key guess: $\hat{r}(p,k_g) = \hat{\rho}(L(y(p,k_g)), \hat{model}_Y(y(p,k_g)))$. The right guess shows the best correlation. In absence of a precomputed model, the attacker splits the trace set and uses one of the parts to build a profile on the fly.

- **Template Attacks** [CRR02]. Using the profiling traces, the attacker estimates the model as the average and covariance of the leak for each of the values of y. Assuming a Gaussian distribution, the attacker can compute the conditional probability density function of L given Y (i.e., $pdf(L=l|Y=y)$) for any of the values of y. Since it estimates also the covariance, this model can capture second order statistical relations. To reduce the complexity, the attacker can first reduce each trace to the POIs only. In the attack phase, the attacker is given a trace l with known plaintext p and unknown key $\tilde{k}$. For each possible guess k_g of the key, the attacker computes the guessed leak variable y_g. The conditional probability density function $pdf(L=l|Y=y_g)$ indicated by the profile can be used as a discriminant for the guess. When multiple attack traces l_i are available, the attacker can combine the results as $\prod_i pdf(L=l_i|K=k_g, P=p_i)$ (maximum likelihood). The correct key guess will correspond to the discriminants with the maximum value. The product can be replaced with a sum of logarithms to avoid numerical problems. To improve accuracy and reduce numerical problems, assuming that the covariance is the same for each value of y, the attacker can compute the pooled covariance matrix as the average of the individual matrices [CK13, OM07, OP11, BJh15].

Profiles are also very useful to analyze the properties of the leaks, for example to study their nonlinearity and to estimate the number of traces required for an attack:

- **Studying Distortion with Linear Regression**. Linear regression was proposed in [SLP05] to fit the leakage function. Naturally, this approach works well when fitting a linear leak with a linear model. By choosing a higher-order polynomial as model, it can also fit more complex (non-linear) leaks. As highlighted in [RSV+11], the non-linear distortion of the leak (compared to the classic Hamming Weight model) can be studied by analyzing the results of linear regression.

- **Fast Security Evaluations** [DFS19]. The attacker estimates the number of traces required for key recovery with Correlation Power Analysis (CPA) from the value of the correlation.

1.5 Studying and Improving the Portability of Profiles

Much like the calibration of a measurement instrument, profiling is affected by a number of factors (e.g., process variability, misalignment, setup) that limit its window of applicability: special care should be taken to compare profiles and to improve portability ([SPRQ06, SKS09, RSV+11, EG12, MBTL13, LPR13, HOTM14, CK14, WO15, CK18, CCC+19, DGD+19, GDD+19]). In the following, we recall those approaches that address the problem at the level of trace/profile normalization. This is effective for our profiled correlation attacks, with traces that are well aligned, sampled at low frequency, and with few POIs that can be easily spotted with the ρ-test. We also discuss how to evaluate profile reuse.

- **Trace Normalization.** An effective way to counter the effect of variability is to use z-score normalization, so that both profile and attack sets have zero mean and unit variance for each POI [MBTL13, EG12]. This is in line with the observation that the main difference among profiles is often only an offset [CK14]. The main disadvantage of z-score normalization is that the average and standard deviation are hard to estimate on a small number of attack traces. For this reason, it might be better to normalize each trace individually by subtracting it with its average value [HOTM14, CK14], or to normalize the attack set using the mean and standard deviation estimated on the profile set [HOTM14].

- **Profile Reuse Metric.** For profiled correlation attacks, the linear correlation $r(P_2,P_1)$ of a profile P_2 with a profile P_1 (built in different conditions) gives the attacker a quantitative measure of the loss introduced by using P_2 instead of P_1. Indeed, the correlation of L_a with P_2 can be computed from the correlation with P_1 as $r(P_2,L_a) = r(P_2,P_1)r(P_1,L_a)$. As a rule of thumb, the number N of attack traces required for an attack to succeed is $N \propto c/\rho^2(P_2,L_a)$, where c is a constant [SPRQ06].

Note on Correlation Metrics. For clarity, we summarize the three cases in which we use correlation metrics, and their meaning. The success of a profiled correlation attack depends on the correlation between the attack traces and those predicted by the profile: $r(L_a,P_1)$. We do not compute this value directly, and we rather provide the key rank for a given number of attack traces. To measure the quality of a profiling set, the ρ-test computes the correlation between a subset of the profiling traces and a profile built on-the-fly with the remaining traces. If the attack traces and the profiling traces were collected in similar conditions, the correlation computed with the ρ-test is a good approximation of $r(L_a,P_1)$. If a profile P_2 was built in different conditions, the correlation with the attack traces which determines the attack success is $r(P_2,L_a) = r(P_2,P_1)r(P_1,L_a)$. Therefore, the correlation between the two profiles $r(P_2,P_1)$ is a good measure of the loss introduced by using a different profile. It can as well be seen as a measure of the similarity/distortion between the two profiles.

1.6 Rank Estimation and Key Enumeration

Typical divide and conquer attacks assign a probability (or score) to each possible value of each sub-key k. Starting from this output, key-ranking algorithms (e.g., [VGS13, YEM14, BLvV15, GGP+15, MOOS15]) estimate how many bits of the key were not recovered, and key-enumeration algorithms (e.g., [VGRS12, BKM+15, MOOS15, PSG16]) bruteforce those bits to recover the key, which can be faster than acquiring more traces.

For example, [PSG16] is based on the idea that a histogram describing which keys fall in a given probability range can be built by convoluting the histograms relative to each sub-key, directly constructed from the probability lists returned by the attack. This histogram is then used to count how many keys precede the right one (key ranking), or to try keys starting from the most likely until the good one is found (key enumeration).

1.7 Basics of Electromagnetic Propagation

The behavior of the electromagnetic field radiated by an antenna changes with distance. As the distance increases, the field quickly moves from being dominantly reactive (reactive near field) to dominantly radiating (radiating near field). Starting from a sufficiently large distance, the field approximately propagates with plane waves (far field). Given an antenna with maximum dimension D, and a field with wavelength λ, the far field approximation is valid at distance d if the following conditions are met: $d > 2D^2/\lambda$, $d >> D$, and $d >> \lambda$. In the far field, at best, the radiated power decreases with the square law of the distance $received\ power/transmitted\ power \sim G_{tx}G_{rx}(\lambda/2\pi d)^2$. G_{tx} and G_{rx} are the gains of the transmitting and receiving antennas, respectively. Higher wavelengths (lower frequencies) propagate better. Sometimes in our experiments we use a coaxial cable to connect the target device with the radio receiver that we use to collect data. In this case the decrease is exponential $e^{-\gamma d}$, where γ depend on the frequency and on the type of cable.

1.8 Diversity Schemes and Multi Channel Attacks

Radio communications are negatively impacted by the noise of the channel. Besides thermal noise, real environments are often subject to deep fade of the signal, for example because of multiple reflections on obstacles along the line of sight between the transmitter and the receiver. Diversity schemes overcome this problem by combining multiple copies of the same original signal, received through independent channels, obtaining a signal increase called diversity gain. In the time domain, diversity can be achieved by transmitting multiple copies of the same message with a period bigger than the channel coherence time. Multiple receivers placed in different locations collect signals that went through different paths (e.g., due to different reflections), or with different polarization, leading to spatial or polarization diversity. Similarly, frequency diversity can be obtained by using different independent portions of the spectrum. There exist several techniques to combine the copies of the signal, for example, selection, equal gain, and maximal ratio (analyzed in [Bre59]). Selection simply consists in choosing the signal with the best quality. Equal gain adds the copies using the same weight, whereas maximal ratio weights the copies proportionally to their quality. Note that the signals must be synchronized before addition.

A similar idea exists in the context of side-channel attacks. Indeed, the same leak variable propagates through a number of different side channels (e.g., power, timing, EM at different frequencies). These multiple copies can be combined to build more efficient multi channel attacks, introduced in [ARR03].

1.9 Overview of the Experimental Setup

For our experiments and attacks, we have significantly extended the open-source framework [Gro18] that we had previously made available for Screaming Channels.

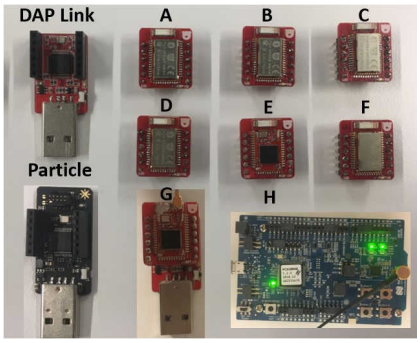


Figure 1: Several instances of the device under attack.

- **Trace Collection.** While we keep the Screaming Channels approach to collect traces in batches (for performance and time diversity), we allow storing all traces of a batch. We now support space diversity with two antennas.
- **Preprocessing.** We have improved the way *AES-128* traces are extracted by improving the computation of the threshold for the trigger. We also added trace normalization (that can also be seen as channel estimation). Details are in Section 2.4.4.
- **Analyses and Attacks.** We have completely rewritten the analysis code in order to implement detection, profiling, comparison and attack based on the k-fold ρ-test. We have also added linear regression, and t-test detection (fixed vs. fixed). Finally, we have increased the speed of the Histogram Enumeration Library [PSG16] with the use of the `Intel AES-NI` instruction, and we have written python bindings to integrate it with our attacks and any other project that may need it.
- **Realistic Environments.** We collect our traces at different distances either in a home environment or in office environment. A simplified scheme to describe these settings is shown in Figure 20b and Figure 20c, respectively. They are two examples of the same type of real environment, but with different geometries and maximum length.
- **Conventional Setup.** We have extended the code and the hardware setup in order to collect traces at the clock frequency, as done by many conventional electromagnetic side-channel attacks. This lets us compare screaming-channel leaks with conventional ones.
- **Real Application.** We have also added a mode that allows collecting traces by interacting with a real authentication protocol (Google Eddystone), as explained in Section 4.
- **Several Instances.** Similarly to Screaming Channels, our Device Under Test (DUT) is based on the *Nordic Semiconductor nRF52832* mixed-signal chip. We use several instances of the *BLE Nano v2* [Red] with the *Particle Debugger* [Par] (*devices A, B, C, D, E, F*, and with the *DAPLink Board* (*devices* F_{DAP}, *G*). We removed the shield of *device E* to place a conventional probe, and we modified *device G* to replace the antenna with a direct cable connection to the radio. For the real application, we also use a *PCA10040* [Norb] evaluation board for the *nRF52832* chip (*device H*). It has an antenna that can be bypassed with a switch to connect to the radio output with a cable. Figure 1 shows these devices.

We use the same *nRF5* [Nora] Software Development Kit (SDK) as Screaming Channels. For reception, we use the following off-the-shelf Software Defined Radios (SDRs):

HackRF [Gre17], *USRP N210* [Etta], and *USRP B210* [Ettb]. To collect the signal, we use off-the-shelf WiFi antennas (standard or directional [TP-]), a custom cable connection, or a custom loop probe based on [emp]. When necessary we use low-noise amplifiers (ZEL-1724LN [Mina], ZKL-1R5 [Minb], ZX60-272LN [Minc]). When not stated otherwise, we attack the `tinyAES` implementation of *AES-128*, with optimization level 0, we tune at 2.528 GHz, and we sample at 5 MHz. The output power of the radio is set to 4 dBm.

The traces previously made available by Screaming Channels were collected at 10 m in an anechoic chamber. We also collect a set of traces at 5 m in similar conditions.

2 Analysis of the Leak Channel

Numerous questions are still open about the nature and properties of screaming-channel leaks compared to conventional side channels. A better understanding of the novel vector is necessary for both defensive and offensive research. In this section we conduct a thorough investigation of the leak channel, supported by extensive experimentation. In this phase, we do not aim at mounting successful and realistic attacks, but at uncovering the characteristics of the leak. Nevertheless, we often use attacks as an analysis tool.

2.1 Open Questions about an Unexplored Channel

One of the prominent features of the novel leak discovered in Screaming Channels is that attacks are possible at considerable distances, because the leak is transmitted by a regular radio front-end. While the advantages of this amplified transmission are intuitively clear, many aspects of the interaction between the leak source and the radio channel are left unexplored.

2.1.1 Coexistence of Intended and Unintended Signals

The coexistence between the leak signals and the signals for the intended radio communication is a first important factor to analyze. In the ideal case, the radio front-end continuously transmits a carrier, which is modulated by the leak. In reality, a modern radio protocol sends data in the form of digitally modulated packets at discrete time intervals. The (absence of) orthogonality between the intended digital modulation and the unintended analog modulation of the leak has a huge impact on how the attacker can recover sensitive information from the radio signal. The radio channel is open intermittently, only when a packet is under transmission. The attacker has to pay great attention to the (absence of) synchronization between sensitive operations and radio packets. If the protocol uses Frequency Hopping Spread Spectrum (FHSS), the packets regularly switch channel (frequency) following a pseudo-random pattern (possibly unknown to the attacker), further increasing the complexity of the reception.

2.1.2 From Digital Logic to the Radio Spectrum

Before being sent over the radio channel, the leak generated by the digital logic has first to flow to the analog/RF part, modulate the carrier, and be amplified and radiated. On the contrary, conventional leaks emanate directly from the digital part. The additional steps taken by screaming-channel leaks involve going through filtering stages and nonlinear components. As a consequence, some of the original information might be lost in the process, countering the advantage of amplification and transmission. Similarly, the shape of the leak traces might encounter considerable distortions compared to conventional ones. The attacker has to understand the impact of such distortions on the leak model, and take it into account to maximize the information that can be extracted.

2.1.3 The Radio Link

Once upconverted and amplified, the leak is radiated by the antenna and it propagates in space. On the one hand, we can imagine that the transmission properties of this leak are rather similar to those of the intended packets. For example, being in the same frequency range, they should have the same type of attenuation and reflection. On the other hand, the leak has some unique properties. First of all, it might not have the same modulation scheme as the packets, resulting in worse resilience to noise. Second, despite the powerful amplification stage, the leak is considerably weaker than the packets, as the input signal is itself weak. Finally, the frequency and bandwidth of the leak might differ from those of the packets, changing the requirements for reception, and potentially conflicting with other channels or protocols. It is important to model the radio channel of the leak, and to understand its attenuation and the distortion it might introduce. The use of time, spatial, or frequency diversity is likely to have a great impact on screaming-channel leaks, since they travel on a long radio link in a possibly complex environment.

2.1.4 Profile Reuse

One of the main advantages of screaming-channel attacks is that they can be mounted from a considerable distance. In this case, the attacker is not likely to have access to the target device. At the same time, profiling the target device at large distance and in challenging conditions might be complex and require a large number of traces. It is therefore necessary to evaluate whether profiles built in certain conditions for a certain device can be reused in different conditions for another instance of the device. As a result, the attacker should be able to take into account the effect of the setup, of distance, of channel frequency, and of the target device. This is potentially more complex than simply trying to reuse profiles on different instances but in the same conditions. While considerable previous work exist on inter-device and inter-setup variations, the effect of distance on profile reuse for screaming channels was never studied before. If possible, reusing a profile taken in convenient conditions on a different instance in more challenging conditions would be a huge advantage for the attacker.

2.2 Coexistence of Intended and Unintended Signals

In [CPM+18], we highlighted that the noise from the digital side of a mixed-signal chips can affect the radio transmitter in many ways. However, we focused on the specific case of a Bluetooth 5 [SIG16] capable device [Red]. In the following, we discuss this scenario in detail, while also reasoning about other similar cases.

2.2.1 Orthogonality of the Modulation

The leak observed on the target device is modulated with analog Amplitude Modulation (AM) on additional carriers that appear at multiples of the clock frequency from the center frequency of the radio channel [CPM+18]. Intended data symbols are modulated with Gaussian Frequency Shift Keying (GFSK) and transmitted at 4 dBm with a modulation rate of 1 Mbps, following the Bluetooth Low Energy (BLE) specification. GFSK is a constant envelope modulation scheme, in which the two digital symbols 1 are 0 are represented as two different frequency offsets of the carrier from its center frequency. Differently from a simple Binary Frequency Shift Keying (BFSK), the transitions between symbols are smoothed with a Gaussian filter, in order to reduce the occupied bandwidth. Since AM is orthogonal to GFSK, the indented receiver will ignore the additional AM leak trace, whereas the attacker will be able to ignore the intended GFSK data. On the one hand, this situation is very convenient for the attacker, since extracting leak traces is easy with a simple Quadrature Amplitude Demodulator tuned at the BLE channel frequency plus a multiple of the clock frequency. On the other hand, analog AM

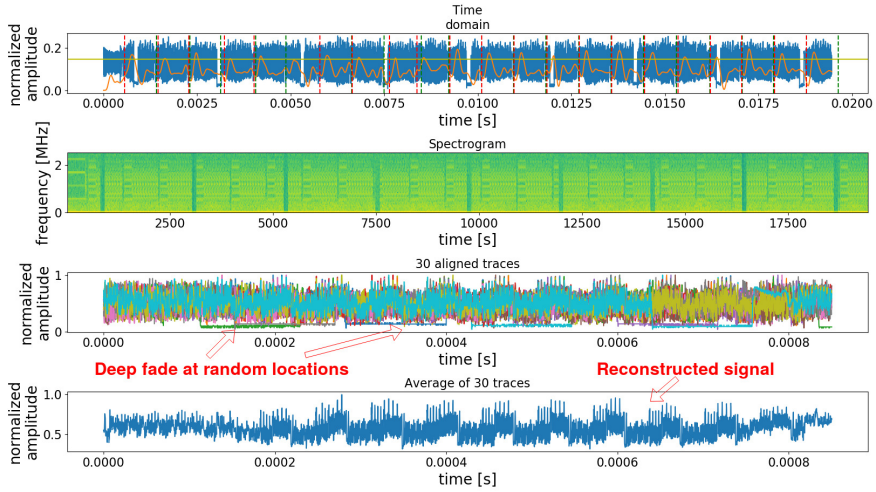


Figure 2: Screaming Channels preprocessing as a form of time diversity: averaging reconstructs the `tinyAES` trace despite the holes between packets.

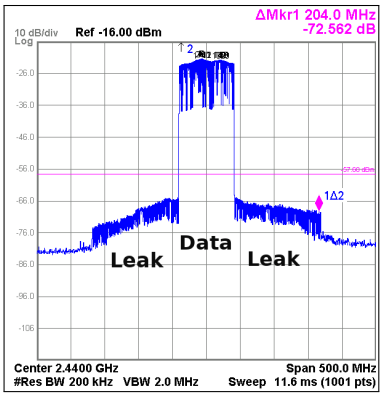
is by far less resistant to noise than GFSK, and the leak does not propagate as well as the intended data. Note that GFSK is a popular choice for many wireless protocols other than BLE, including Bluetooth, Adaptive Network Topology (ANT)/ANT+ [ant], Gazel™ [Semb], and Enhanced ShockBurst™ [Sema], which are all supported by our target chip.

2.2.2 Discrete Packets as Deep Fade

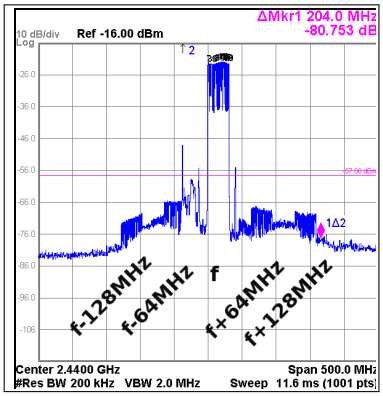
The firmware used in Screaming Channels [Gro18] configures packets to have a random address and a random 254-byte payload. We observe that this results in packets of ≈2.1 ms, spaced by periods of ≈120 µs during which the radio transmitter is off. Encryptions are ≈820 µs long, spaced by ≈48 µs, and they are not synchronous with the packets. As a result, from the point of view of the leak, the channel is intermittently off. The leak traces collected by the attacker might contain deep holes of ≈120 µs, corresponding to the periods when the radio is not on. This phenomenon, that we can interpret as deep fade, is a peculiar characteristic of screaming-channel leaks compared to conventional ones. Nevertheless, we can imagine specific cases in which the attacker might (at least partially) synchronize encryptions and packets. For example, the hardware block could be configured to encrypt packets on-the-fly during transmission, or the attacker could trigger an encryption just before a transmission. In this section, we are interested in studying the channel independently of specific attacks, and we use asynchronous packets and encryptions as a general case for our investigation. We inherit from Screaming Channels a preprocessing technique consisting in averaging many leak traces corresponding to the same encryption. In this context, it can be interpreted as a form of time diversity to reconstruct a full trace despite random holes in the signal. Deep fade due to spaces between packets, and trace reconstruction using time diversity, are visible in Figure 2.

2.2.3 Frequency Hopping

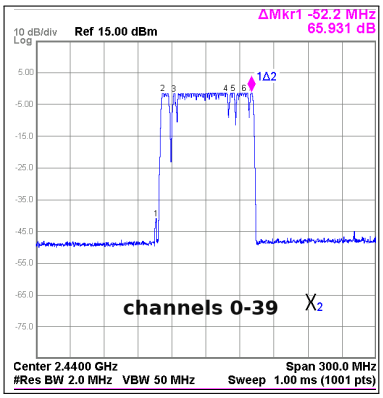
Except for the real world attack in Section 4, frequency hopping is disabled and the carrier is fixed at 2.4 GHz. As in Screaming Channels, we assume that a motivated attacker could build a receiver able to follow the hopping sequence, listen to all channels simultaneously with a wider band or multiple receivers, or listen at a fixed frequency and wait for the carrier to jump to that channel. The last case might be particularly reasonable for the advertising



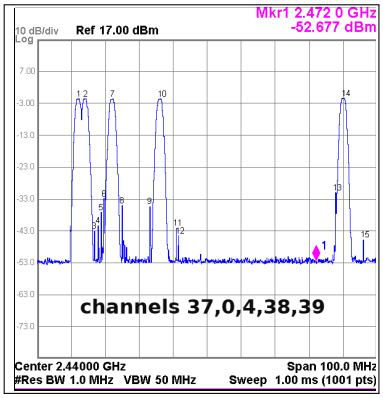
(a) Test mode, 40 channels, data and leak



(b) Test mode, 20 channels, data and leak



(c) Real connection, 3 advertising and 37 data channels (default)



(d) Real connection, 3 advertising and 2 data channels (specified by the master)

Figure 3: To study hopping, we connect *device H* to a spectrum analyzer in maximum hold mode. Each time a packet is sent on a channel a spike appears at the corresponding frequency. In test mode we configure the device to sweep over the BLE channels, data and the leak at the multiples of the clock are clearly visible (a)(b). In a real connection, the device uses all the 40 channels by default (c). We can reduce the channels used by a real connection to 5 by blacklisting the others from the master side (c).

packets, which are sent over only three fixed channels. In addition, under certain conditions the attacker can reduce the number of channels used for hopping. A BLE master can ask the peripheral to blacklist up to 35 data channels out of 37. This is a standard feature of the BLE standard, used to reduce interference by adapting to the environment (adaptive frequency hopping). In practice it can be done by calling a function of the SDK, or by issuing a *Set Host Channel Classification Command* [SIG16] to the BLE dongle that acts as master. In Section 4 we show that being a master and reducing the number of hopping channels is a reasonable assumption in a real-world attack scenario. A non-master attacker could still try to jam some of the channels, hoping that a master that performs channel assessment will blacklist them. Figure 3 shows the leak in presence of hopping (in test mode) and the reduction of the hopping channels for a real connection. On the one side frequency hopping makes the attack more complex, on the other side the transmission of the leak profits from the same advantages of frequency hopping as the regular data. For example, hopping helps to reduce the impact of interference from other transmitters, and other narrow band signals in general. Other GFSK protocols might not use FHSS. For example, the

ANT/ANT+ protocol avoids interference by dividing the channel into time slots. ANT+ devices might change channel at the application layer if they detect that the current one is too crowded (Frequency Agility). Similarly, Enhanced ShockBurst™ does not use FHSS, unless implemented at the application layer. Moreover, we observe that most protocols send an acknowledge packet at reception. An attacker can trigger a transmission (of an acknowledge packet) on a known channel by first sending a packet to the target device. To conclude, assuming a fixed channel is a reasonable choice for the study of the leak vector.

2.2.4 Interference

The frequency of the leak is the BLE channel frequency plus (or minus) a multiple of the 64 MHz clock [CPM+18]. It is reasonable to assume that, for each channel, there exists a choice of the harmonic for which the leak is less vulnerable to interference from other channels and protocols. For example, if we take the BLE channels from 2.402 GHz to 2.480 GHz, and the second harmonic of the clock, the leak will span from 2.530 GHz to 2.608 GHz. In this range, which is well outside the Industrial, Scientific and Medical (ISM) band, the leak is less likely to be degraded by WiFi, BLE, and other 2.4 GHz protocols.

2.3 From Digital Logic to the Radio Spectrum

The path from digital logic to the radio spectrum through the radio front-end is a distinguishing feature of screaming-channel leaks. We want to investigate the effect of this path on the strength of the leak and on its shape. In this phase, we keep the distance small and fixed. We normalize the traces, as described in more detail in Section 2.4.4.

2.3.1 Strength of the Leak

We want to understand whether screaming-channel leaks are as strong as the conventional ones, or some loss appears when the leaks go through the radio transmitter. We are interested in the order of magnitude of the possible difference between screaming-channel and conventional leaks, for comparable setups and number of traces. To this purpose, we collect a set of $5000 \cdot 500$ profile traces and $1500 \cdot 500$ attack traces, for the three following cases:

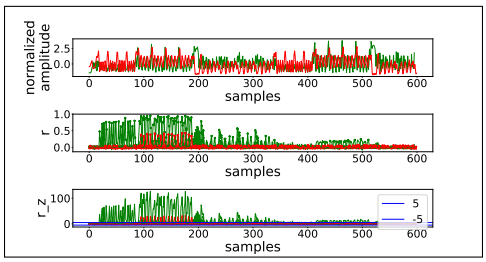
- **Conventional (*device E*).** Traces are collected with a loop probe at the clock frequency (64 MHz). The loop probe is placed close to the power supply pin of a *BLE Nano v2* without metallic shield, where the signal is stronger. Further amplification with a ZKL-1R5 Low Noise Amplifier (LNA) is required.
- **Screaming via Cable (*device G*).** Traces are collected at radio frequency (2.528 GHz), but using a custom cable connection between the radio and a *BLE Nano v2* modified for this purpose.
- **Screaming at 10 cm (*device E*).** Traces are collected at radio frequency (2.528 GHz), using a standard WiFi antenna at a distance of 10 cm from a *BLE Nano v2*, in a home environment. The radio is a *HackRF*. Note that this case is representative of a simple realistic screaming-channel attack, and it is comparable with the conventional setting, but it not necessarily the optimal one.

For each case we carefully optimize the collection parameters. We use a HackRF radio and a sampling rate of 5 MHz.

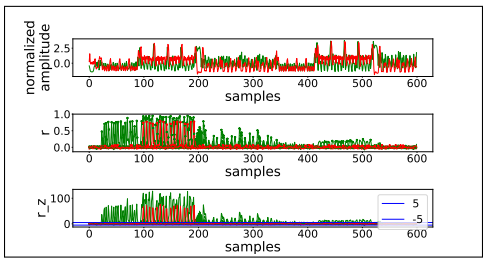
We then evaluate the ρ-test. Results are shown in Figure 4. In the conventional setting, several POIs emerge for each byte of the key, and correlation is high. On the contrary, correlation decreases for the screaming-channel settings, and only one POI emerges for each byte. The correlation for the cable connection is lower, probably because the load for transmitter is not optimal (we replaced the antenna with a custom connection via cable).

Table 1: Summary of the attacks in the conventional, screaming-channel via cable, and screaming-channel at 10 cm settings. All trace numbers should be multiplied by 500. p.c. stands for profiled correlation, and t.a.p.c. stands for template attack with pooled covariance. The rank was computed with enumeration. The higher correlation and number of POIs result in better attacks for the conventional setting, as expected.

	Profile	max ρ,r_z	Attack	Type, POIs	Variable	Rank
Conventional	$5k$	0.95, 128	3	t.a.p.c., 11	$p \oplus k$	2^{18}
Conventional	$5k$	0.95, 128	5	t.a.p.c., 11	$p \oplus k$	0
Conventional	$5k$	0.95, 128	9	t.a.p.c., 1	$p \oplus k$	2^{16}
Conventional	$5k$	0.95, 128	9	p.c., 1	$p \oplus k$	2^{11}
Conventional	$5k$	0.95, 128	14	t.a.p.c., 1	$p \oplus k$	0
Conventional	$5k$	0.95, 128	14	p.c., 1	$p \oplus k$	0
10 cm	$5k$	0.79, 76	130	t.a.p.c., 1	$p \oplus k$	2^{21}
10 cm	$5k$	0.79, 76	130	p.c., 1	$p \oplus k$	2^{21}
10 cm	$5k$	0.79, 76	1273	t.a.p.c., 1	$p \oplus k$	2^{8}
10 cm	$5k$	0.79, 76	1273	p.c., 1	$p \oplus k$	0
Cable	$5k$	0.43, 32	1500	p.c., 1	$p \oplus k$	$> 2^{21}$



(a) Screaming Cable (red) vs. Conventional, ρ-test



(b) Screaming at 10 cm (red) vs. Conventional, ρ-test

Figure 4: Results of the ρ-test (average trace, correlation, confidence). Comparison between screaming-channel via cable and conventional (a), and between screaming-channel at 10 cm and conventional (b). Correlation peaks and number of POIs are higher for the conventional case.

The correlation at 10 cm is on the same order of magnitude than the conventional one, though smaller. Better results for correlation and number of POIs can be achieved in better settings, as shown throughout the paper (e.g., in Section 2.4.5 and Section 2.6.2).

To give a more practical metric, we compute the number of attack traces necessary to recover the key using profiled attacks and key enumeration, as shown in Table 1. Conventional attacks are more efficient because of the higher correlation and because of the higher number of available POIs.

It is also interesting to compare profiled correlation attacks and univariate template attacks. As explained in detail in Section 1.3, both profiled correlation and templates can capture the nonlinearity of the leak model, thanks to the profiling step which estimates the model for every possible value of $p \oplus k$. However, template attacks can capture a second order relation between the leak and the model, whereas profiled correlation attacks can only capture linear correlation. For both the conventional and screaming-channel cases, template attacks are not more efficient. We can then assume that, for our sample size, correlation is good enough, and there is no significant advantage in using methods that can capture higher order statistics, such as templates and mutual information. Additionally, profiled correlation attacks are computationally less expensive.

2.3.2 Distortion

We want to understand whether screaming-channel leaks follow the same leak model as conventional ones, or are distorted on the way to the radio link. This is both interesting from a fundamental point of view, and useful to improve the attacks by choosing the best leak model. We reuse the same data as for the strength of the leak (Section 2.3.1).

Conventional Leaks are Not Distorted. The conventional leak follows very closely a simple Hamming Weight model, and the highest POIs correspond to the output of the first Sbox. The ρ-test shows very similar correlation at the same point both for $p \oplus k$ and $HW[Sbox[p \oplus k]]$ (Figure 5). This confirms the point as the output of the Sbox, and it shows that a linear leak model is a good assumption. As an additional check, we compute the non-profiled correlation $\hat{r}(L,Y)$ with $y = HW[Sbox[p \oplus k]]$, and we observe that a similar (negative) correlation peak appears: $\hat{r}(L,Y) = -0.9$, $-10log(p) > 20$. This matches with the profile built before.

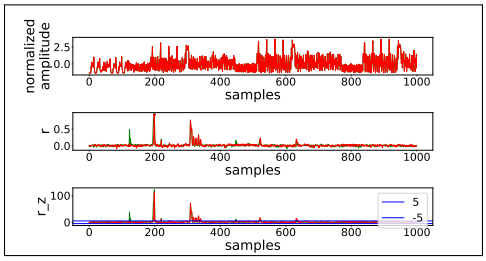
Screaming Leaks are Distorted. On the contrary, the screaming-channel leaks have a distorted profile, both via cable and at 10 cm. In these cases, choosing $y = HW[Sbox[p \oplus k]]$ significantly reduces the correlation peaks of the ρ-test compared to templates built for $y = Sbox(p \oplus k)$ or $y = p \oplus k$, because the Hamming Weight model is not a good assumption. If we compare the profile built for $y = p \oplus k$ for the conventional and the screaming-channel case, they are not correlated due to the distortion of the latter. Instead, screaming-channel profiles for cable and 10 cm show a correlation of 0.59 ($-log10(p) = 25$) between each other. This is clearly visible in Figure 6, together with the shape of the distorted profile. Results are similar for the other bytes of the key. Table 2 shows the impact on the attacks.

We further study this distortion using linear regression. In particular, we want to understand whether a linear combination of the bits of the output of the Sbox is enough to model the leak better than the Hamming Weight, or if the leak is non linear. First, we choose $y = Sbox(p \oplus k)$, and we find the parameters A and B for which $\hat{model}_{linear}(y) = A + B \cdot Y$ best fits the measured leak L, where Y is the vector of the bits of y. While still linear, this model can take into account different weights for the bits, whereas the Hamming Weight model considers them all equal. Then, we remove the assumption on linearity, and we estimate the profile directly for each value of y: $\hat{model}_{full}(y) = \bar{L}_y$ (as we do for the ρ-test). For the conventional case, the linear fit is a good approximation, and $\hat{model}_{linear}$ has a correlation of 0.91 with $\hat{model}_{full}$ ($-log10(p) = 984$). For the screaming-channel case, the linear fit is not a good approximation, and the $\hat{model}_{linear}$ has a correlation with $\hat{model}_{full}$ of only 0.17 (($-log10(p) = 22$). This is visible in Figure 7. Results are similar for the other bytes. We conclude that, for the screaming-channel case, the relation between the bits of the output of the Sbox and the leak is not linear.

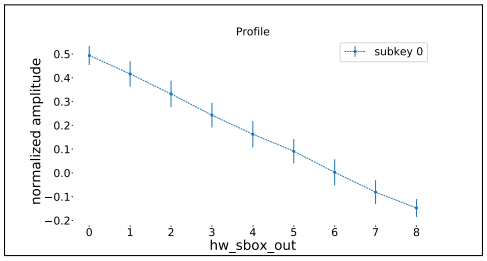
Since it appears only for screaming channels, we conjecture that the nonlinearity of the model is due to the path between the digital logic and the radio transmitter. Estimating the profile for the 256 values of $y = Sbox(p \oplus k)$ (or of $y = p \oplus k$) can capture the nonlinearity of the model, and it is the best strategy for analysis and attack.

2.3.3 The Impact of The Channel Frequency

Profiles built at different frequencies might differ because of the different frequency behavior of the components along the path that brings the leak to the antenna. Our target device with our firmware can transmit on 81 channels from 2.400 GHz to 2.480 GHz, spaced by 1 MHz. The advertising channels of BLE are particularly interesting for attacks because they are at a fixed known frequency. We decide to analyze the frequency used by Screaming Channels (2.400 GHz), and the advertising channels: channel 37 (2.402 GHz), channel 38 (2.426 GHz), and channel 39 (2.480 MHz). For extraction, we tune at $f_{channel} + 2 \cdot f_{clk} = f_{channel} + 2 \cdot 64$MHz like Screaming Channels. Using a *USRP B210* with a standard WiFi antenna at 10 cm from a *BLENano v2* (*device E*) in an office environment, we collect $5000 \cdot 500$ profiling traces and $2000 \cdot 500$ attack traces, for each of the channels.

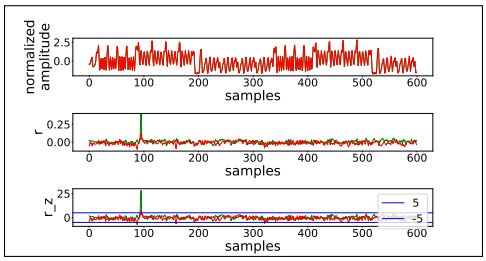


(a) ρ-test with $p \oplus k$ (green) and $HW(Sbox(p \oplus k))$

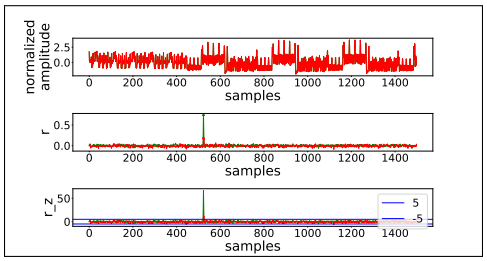


(b) Profile for $y = HW(Sbox(p \oplus k))$

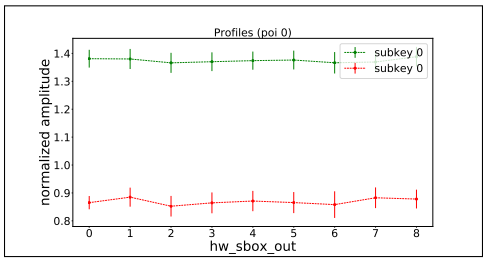
Figure 5: In the conventional case the Hamming Weight model is a good assumption. Results of the ρ-test for the first byte of the key (a). Assuming the $HW[Sbox[p \oplus k]]$ leak model (red) does not significantly reduce correlation for the highest POI, compared to choosing $p \oplus k$ (green). The profile (b) clearly shows almost linear negative correlation.



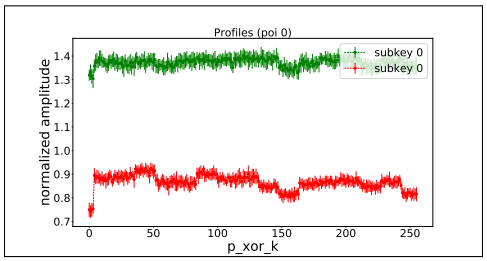
(a) Screaming Cable: ρ-test with $p \oplus k$ (green) and $HW(Sbox(p \oplus k))$ (red)



(b) Screaming 10 cm: ρ-test with $p \oplus k$ (green) and $HW(Sbox(p \oplus k))$ (red)



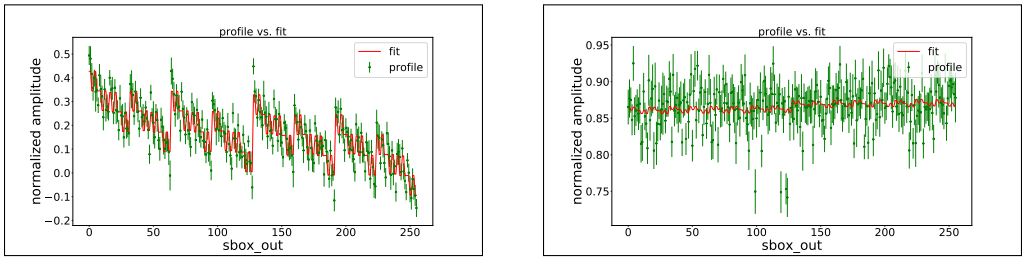
(c) Profile built with $HW(Sbox(p \oplus k))$ for cable (green) and 10 cm (red)



(d) Profile built with $p \oplus k$ for cable (green) and 10 cm (red)

Figure 6: Screaming channels leaks follow a distorted leak model. For both the connection via cable (a) and at 10 cm (b), choosing $HW(Sbox(p \oplus k))$ (red) drastically decreases the ρ-test correlation compared to choosing $Sbox(p \oplus k)$ or $p \oplus k$ (green). Profiles for cable and 10 cm (c)(d) are similar, but they differ from the conventional case.

Channel Frequency Does Not Impact Distortion. For each set we run the ρ-test, and we correlate each profile with the one at 2.400 GHz to observe if there is any distortion. Results in Table 3 show that the amount of correlation and the shape of profiles is mostly constant. We conclude that the distortion of screaming channels does not have a strong dependency on the frequency of the channel.



(a) Conventional **(b)** Screaming at 10 cm

Figure 7: For the bits at the output of the Sbox, the linear model (red) is a good approximation of the leak model (green) for the conventional case (a), but not for screaming channels (b).

Table 2: Attack results confirm that choosing $y = p \oplus k$ is important to take into account the distortion present in screaming-channel leaks, whereas it does not make any significant difference in the conventional case. p.c. stands for profiled correlation, whereas t.a.p.c. stands for template attack with pooled covariance.

	Profile	max ρ,r_z	Attack	Type, POIs	Variable	Rank
Conventional	$5k$	0.93, 117	14	t.a.p.c., 1	HW	0
Conventional	$5k$	0.93, 117	14	p.c., 1	HW	0
Conventional	$5k$	0.95, 128	14	p.c., 1	$p \oplus k$	0
Conventional	$5k$	0.95, 128	14	t.a.p.c., 1	$p \oplus k$	0
10 cm	$5k$	0.20, 14	1273	t.a.p.c., 1	HW	$>2^{27}$
10 cm	$5k$	0.20, 14	1273	p.c., 1	HW	$>2^{27}$
10 cm	$5k$	0.79, 76	1273	t.a.p.c., 1	$p \oplus k$	2^8
10 cm	$5k$	0.79, 76	1273	p.c., 1	$p \oplus k$	0

2.4 The Radio Link

Screaming-channel leaks travel over a radio link between the target device and the receiver of the attacker. The distance that can be achieved with this setting is considerable compared to conventional attacks. We want to understand the properties of this channel and to exploit it at its best.

2.4.1 A Simple Model of the Leak Transmission Chain

With a simple model from radio theory, we show the inherent advantage of screaming channels over conventional leaks, especially in case of large attack distance. Moreover, we explain which optimizations of the setup have the largest impact in improving the reception of the leak.

A Simple Model. Figure 8 shows a simple model of the transmission chain between the target and the radio of the attacker. The input of the chain is the leak trace, which has

Table 3: Similarity among profiles ($\hat{r}(P_i, P_2)$) and strength of the leak ($max\rho, r_z$) for different channel frequencies (and the corresponding frequency at which we tune). Profiles are similar, showing that the distortion of screaming-channel leaks is constant with frequency. The same applies for the amount of correlation.

	channel f (GHz)	**tune f (GHz)**	$\hat{\boldsymbol{r}}(\boldsymbol{P_i}, \boldsymbol{P_2})$, **-log10(p)**	$\boldsymbol{max\rho, r_z}$
$\boldsymbol{P_0}$	2.400	2.528	1.00, inf	0.79, 74.73
$\boldsymbol{P_1}$	2.402	2.530	0.83, 66.39	0.79, 75.27
$\boldsymbol{P_2}$	2.426	2.554	0.86, 75.82	0.77, 71.22
$\boldsymbol{P_3}$	2.608	2.480	0.85, 71.32	0.68, 58.49

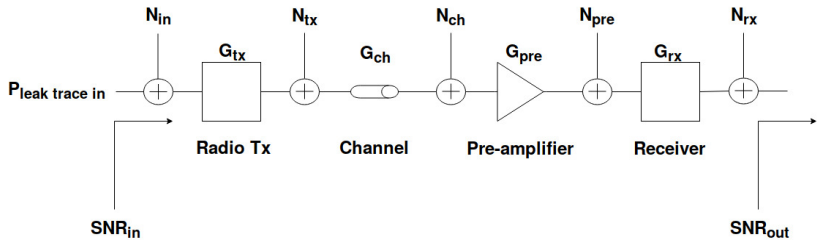


Figure 8: Simple model of the transmission chain between target and attacker.

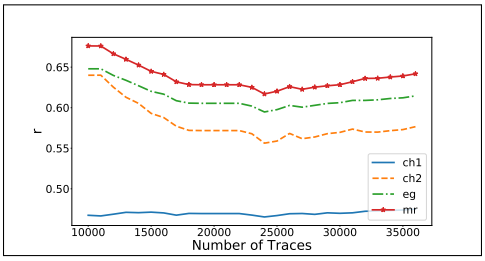
$SNR_{leak\ trace\ in} = P_{leak\ trace\ in}/N_{leak\ trace\ in}$, where the numerator is the power of the leak trace and the denominator is the power of the noise (e.g., thermal noise) over the frequency band of interest. Following [CPM+18], the leak trace is upconverted and amplified by the radio transmitter. Then, it goes through the channel, a pre-amplifier, and finally the receiver. Each block has a gain G and it introduces some additive noise N. The SNR at the output of each block is reduced by a factor called noise factor $F = 1 + \frac{N}{N_{in}G}$. The total noise factor of our chain is $F = F_{tx} + \frac{F_{ch}-1}{G_{tx}} + \frac{F_{pre}-1}{G_{tx}G_{ch}} + \frac{F_{rx}-1}{G_{tx}G_{ch}G_{pre}}$, and the Signal-to-Noise Ratio (SNR) at the output is $SNR_{leak\ trace\ out} = SNR_{leak\ trace\ in}/F$. Note that, at each step of the chain, the gain of the previous stage reduces the impact of the noise of the next stage. Early amplification has dominant positive effects on the SNR compared to amplification in the following stages. As explained in Section 1, the gain[1] of the channel in far field is $G_{ch} = G_{antenna\ tx} G_{antenna\ rx} (\lambda/2\pi d)^2$.

Inherent Advantage of Screaming Channels and Setup Optimization. The two main advantages of screaming-channel leaks over conventional leaks can be clearly spotted in the formula of the noise factor F. First, the trace leak is amplified as early as possible in the chain by the radio transmitter (G_{tx}), reducing the impact of the noise factor of the channel F_{ch}. This is important because F_{ch} increases with the square of the distance d. Second, the intended radio transmitter is likely to have a good antenna gain $G_{antenna\ tx}$, which also contributes to reduce the total noise factor. As the intended transmitter block is not present in the case of conventional leaks, these advantages cannot be exploited for long range attacks. From the formula for F we can also observe the advantage of two well-known optimizations of the setup. First, the attacker can choose a directional antenna with a high gain $G_{antenna\ rx}$. Second, the attacker can use a low noise amplifier with a good noise factor to preamplify the traces before feeding them to the radio. The next amplification stages have less impact on the noise factor of the analog part, but they are important to match the trace amplitude with the dynamic range of the analog-to-digital converter (ADC) to avoid losses in resolution.

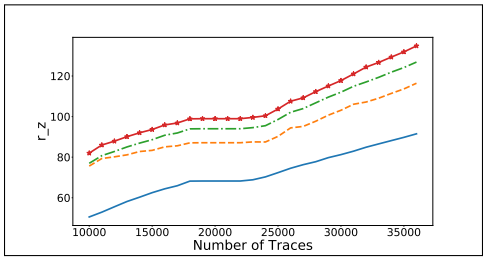
2.4.2 A More Complex Channel

The radio channel between target and attacker is not affected by thermal noise only. First, as seen in Section 2.2, a screaming channel is inherently characterized by intermittent deep fade. It is not active when packets are not transmitted, and leak traces are not synchronous with the packets. Second, a number of other non ideal effects must be taken into account. The difference between the clocks of the receiver and transmitter will produce time-varying frequency offsets that impact demodulation. The channel introduces distortions because different frequencies in the band of interest undergo different losses. In real environments, the signal encounters obstacles that produce reflections. The resulting multiple paths have different delays and losses. Finally, gains and losses are different for different distances and setups, and they are not necessarily constant with time and temperature. These problems should be addressed during reception and preprocessing. Although at least some of these

[1] Note that the gain is less than 1 so it is actually a "loss".



(a) ρ vs. Number of Traces



(b) r_z vs. Number of Traces

Figure 9: Comparison among spatial diversity and single channels. Maximal ratio and equal gain show a net advantage in terms of correlation over channel 1 and channel 2 for any number of traces.

issues are typical of any channel, including the conventional one, they are particularly important for radio communications such as screaming-channel leaks.

2.4.3 Time, Spatial, and Frequency Diversity

In [CPM+18], we did not use any form of preprocessing, apart from producing each trace $l_{\bar{m},y_{p,k}}(t)$ as the average of m traces with the same plaintext/key. As explained in Section 2.2, we observe that this technique is a form of time diversity. The same information ($l_{y_{p,k}}(t)$) is transmitted m times over a noisy channel. The average of these copies helps to reduce the effect of any type of uncorrelated noise. This is true at least for thermal noise and for the deep fade at random locations due to the lack of synchronization between packets and leak traces.

We can further improve the quality of the leak traces using multiple spatial and frequency streams. Differently from time diversity, they increase the complexity and cost of the setup, but they do not require the target to repeat multiple times the same operation. Frequency diversity is possible because the leak is broadcast at multiple harmonics, as explained in Screaming Channels. Since these harmonics are spaced by 64 MHz, they are likely to be impacted by uncorrelated noise. For example, an interfering signal will not have a bandwidth large enough to impact two harmonics at the same time. However, this large spacing is also more demanding for the receiver, and impossible with our current experimental setup. For this reason, we leave it for future work, and we focus on spatial diversity.

Improvements With Spatial Diversity. We use two receive chains of a *USRP B210* radio to receive the leak traces from two antennas at different positions. The leak traces might arrive at the two antennas with different delays, because they go through different paths. However, they are transparently synchronized by the extraction process. Once aligned, the two traces coming from the two antennas can be combined together. We implement both equal gain and maximal ratio combination techniques. For the latter, we use a simple way of computing the quality of a trace: $\bar{l}(t)/std(l(t))$. To compare the results, we collect a set of 37000·500 profiling traces at 10 cm in a home environment (*device E*). We use two standard WiFi antennas spaced by 3 cm. Figure 9 shows the maximum correlation (and confidence) for a varying number of traces. The advantage of maximal ratio and equal gain over each of the single channels is evident. We will show concrete examples of attacks in Section 3.3.

2.4.4 Normalization, Channel Estimation, and Profile Reuse

As seen in Section 1.5, normalization is an important method to improve the portability of profiles built in different conditions (e.g., device instance, setup, time). The radio channel is an additional source of variability for screaming channels. The total gain of the channel G (including the setup) changes with distance and it might not be stable over time. Different

distances also force the attacker to use different equipment, increasing the variability of the setup. Normalization has to take into account these considerations.

In Screaming Channels, traces are not normalized. As a consequence, profiles cannot be reused across different settings. To solve this problem, in the code release we suggested to divide each trace by its average ([Gro18]). We observe that this accomplishes both normalization of the mean and channel estimation. The original leak trace $l_{tx}(t)$ undergoes several stages of gain/loss $l_{rx}(t) = l_{tx}(t)G_{total}$. If we assume that the gains are constant during the short duration of the leak traces, the average is $\bar{l}_{rx} = \bar{l}_{tx}(t)G_{total}$, and the normalized trace is $l_{tx}(t)/\bar{l}_{tx}$. If the gains are different for different traces (e.g., because of a slow drift due to temperature, or different setups), the resulting noise is filtered out as well.

Normalization and Channel Estimation. Building on previous work ([HOTM14, CK14, MBTL13, EG12]), we propose a more complete form of normalization. Instead of applying z-score normalization to the entire set, or to remove the DC offset from each trace, we apply z-score normalization to each trace. Assuming a normal distribution, each point of the resulting traces will belong to $\mathcal{N}(\mu = 0, \sigma = 1)$. The advantage of this method is that it works as a form of per-trace channel estimation, which can filter out the variations of gain with time, temperature, and other external factors. This is particularly important for screaming channels traces, because an acquisition campaign might be long and the environment might be subject to change. Under the same assumptions as before, $\bar{l}_{rx} = G_{total}\bar{l}_{tx}$, $std(l_{rx}) = G_{total}\,std(l_{tx})$, and the normalized trace is $\frac{l_{tx}(t) - \bar{l}_{tx}}{std(l_{tx})}$. Independently from the conditions in which the leak was transmitted and received, all normalized traces are comparable. This improves the portability of profiles, as explained in Section 2.5.

Additional Preprocessing Techniques. As of now, this normalization is applied after having extracted a single trace from m traces. We leave as future work the investigation of better preprocessing techniques, including normalization, to apply to each trace before averaging. For example, we could exclude those points that are affected by deep fade, and remove noise components with filters in the frequency domain or with Singular Spectrum Analysis (SSA) [GZ13, VYG92, PS15]. As of now alignment is mainly based on cross-correlation, we could instead use Dynamic Time Warping (DTW) [vWWB11] to account for instabilities in the clock. We could as well track the frequency offset, and equalize the traces if the channel gain is not constant with frequency.

2.4.5 The Impact of Distance (and Setup) on Correlation and Distortion

In the far-field[2], the main effect of distance is the quadratic reduction of the gain of the channel. To compensate for this loss we have to use higher amplification and higher-quality components. This is why we cannot fully decouple the effect of distance from that of the setup. We optimize reception for each distance we want to study, and then we compare the results. This is a fair and realistic case, since also the attacker will adapt the setup to the conditions. In addition to the traces collected in Section 2.3, we collect the following sets of 5000·500 traces:

- **Home 20 cm (*device E*).** Traces are collected with a standard WiFi antenna and a *HackRF* radio. The target is a *BLE Nano v2*.
- **Office 1 m (*device F*).** Traces are collected with a *USRP N210* radio. To avoid reflections, and to increase the gain, we use a directional antenna with 24 dBi gain ([TP-]), and two low-noise amplifiers with 20 dB gain and 1.5 dB noise figure ([Mina]). The target is a *BLE Nano v2* (with *Particle Debugger*).
- **Anechoic 5 m.** Same as above, but in an anechoic chamber, against a *BLE Nano v2*.

[2] Our leak is at 2.528 GHz and the maximum size of the antenna of the target is 6 mm, which gives the following far field conditions: $d > 2D^2/\lambda = 0.6$mm, $d >> D = 6$mm, and $d >> \lambda = c/f = 119$mm.

- **Anechoic 10 m.** Traces of Screaming Channels available at [Gro18], with the same type of setup as above.

For each set, we run the ρ-test, and we build the profile, choosing $y = p \oplus k$ as leak variable. Then, we measure the similarity among profiles by correlating each profile with the one taken at 10 cm. Results about distortion and correlation are visible in the last two columns of Table 4, respectively.

Correlation Stays High. We observe that with enough amplification it is possible to collect good traces even at large distance. The quality of the setup and of the environment seem to play a bigger role. For example, the amount of correlation at 5 m in an anechoic chamber with a good setup is even higher than correlation at 10 cm in a home environment with a poor setup. An attacker can profit from the best conditions during the profiling phase. Overall, the amount of correlation tends to be high in all cases.

Distortion is Stable. All the profiles built in screaming-channel conditions correlate well among each other. Only the setup via cable shows a lower correlation. On the contrary, the profile built in the conventional setting is not correlated with the screaming ones. This shows that the distortion of the profile is stable with distance and across setups. This confirms the hypothesis made in Section 2.3.2 that distortion happens on the path from the digital logic to the radio channel.

2.5 Profile Reuse

In this paper we focus on profiled correlation attacks, since they are well adapted to screaming-channel traces. As explained in Section 1, we can evaluate profile reuse by computing the correlation between different profiles. We have already observed that, although the profile of the leak might vary with the code layout, it is instead stable with distance, setup, and channel frequency. In the following we will focus on the impact of two additional factors: acquisition campaign and device instance. In this phase, we use 5 *BLENano v2.0* devices (A, B, C, D, E, and F), with a *Particle Debugger*. We also try device F with a *DAPLink Board* (that we call F_{DAP}). All measurements are taken in the same office setup at 10 cm, using a *USRP B210* software-defined radio (SDR), and a standard WiFi antenna.

2.5.1 Reuse Over Time

In real attack scenarios, profiling and attack traces are most likely collected in distinct acquisition campaigns, at different times. The difference between acquisition campaigns cannot be avoided, even when profiling and attacking the same device. Using device A, we collect 8 profiling sets of $5000 \cdot 500$ traces, followed by 8 attack sets of $2000 \cdot 500$ traces. The total collection time is around 22 hours. Results in Table 5 show that the amount of correlation and the

Table 4: Similarity among profiles ($\hat{r}(P_i, P_2)$) and strength of the leak ($max\rho, r_z$) for different distances and setups. All profiles taken for screaming channels are similar, showing that the distortion of screaming-channel leaks is constant. The amount of correlation stays high despite distance, and it rather depends on the quality of setup and environment.

	d (m)	**environment**	**antenna**	$\hat{\boldsymbol{r}}(\boldsymbol{P_i}, \boldsymbol{P_2})$, **-log10(p)**	$\boldsymbol{max\rho, r_z}$
$\boldsymbol{P_0}$	0.00	conventional	loop probe	-0.07, 0.56	0.95, 127.35
$\boldsymbol{P_1}$	0.00	cable	n.a.	0.55, 21.09	0.39, 28.96
$\boldsymbol{P_2}$	0.10	home	standard	1.00, inf	0.79, 75.72
$\boldsymbol{P_3}$	0.20	home	standard	0.96, 142.77	0.77, 72.30
$\boldsymbol{P_4}$	1.00	office	directional	0.40, 10.32	0.41, 30.66
$\boldsymbol{P_5}$	5.00	anechoic	directional	0.96, 139.51	0.85, 89.84
$\boldsymbol{P_6}$	10.00	anechoic	directional	0.92, 107.80	0.77, 71.71

shape of the profile are quite stable for different campaigns despite several hours of difference. In Section 3.4.2, we will show that a profile can be reused several months later as well.

2.5.2 Reuse Across Devices

To evaluate the portability of profiles across devices, we also collect 2 sets of $5000 \cdot 500$ profiling traces and 2 sets of $2000 \cdot 500$ attack traces, for devices B, C, D, E, F, and F_{DAP}. Results in Figure 10 show that profiles taken during different acquisition campaigns and on different devices can be reused without considerable loss. Only a few cases (*B0*, *B1*, *C1*, *E1*) behave as outliers, because the quality of the profile is low (probably due to some external factor during the acquisition in a realistic environment). In this case we repeat the measurement. The leak is clearly visible and in the same order of magnitude in most profiles in Figure 10b. Profiles are very similar to each other, as shown in Figure 10b, where each profile is compared with the one taken on device A. In the few cases when the correlation among profiles is low, the cause is the low quality of one of the profiles rather than a significant difference in their shape. Normalization brings a clear advantage over raw traces. In addition to correlation, we also provide some attack metrics in Figure 11. For each profile, we run an attack with traces collected with the same device or with traces collected on device A. We show the evolution of the key rank with different campaigns and devices. In general, the key rank increases when attacking different devices, but not significantly, and it remains in the same order of magnitude as in the case of different acquisition campaigns. Normalization clearly improves the portability of the templates. The same applies when averaging the profiles of the 16 bytes together, as we will explain more in detail in Section 2.6.3.

2.6 Other Practical Observations

We have investigated a novel channel and uncovered its peculiar properties. We now explore a few additional practical aspects:

- The point at which key enumeration becomes useful given the slow collection speed of screaming-channel traces.
- The impact of the extremely low sampling rate of screaming channels on the number of POIs, and how to best exploit them when there are more than one.
- How to combine the profiles of several bytes for a more accurate estimate.
- The role of the connection between the target device and the attacker in our current experimental setup.

2.6.1 Key Enumeration

In general, side-channel attacks quickly recover most of the bits of the key, but then take many more traces to converge to full recovery. When the convergence rate slows down, it may

Table 5: Comparison among several profiling campaigns.

	$\hat{r}(P_i,P_2)$, -log10(p)	ρ,r_z
P_{A0}	1.00, inf	0.81, 80.70
P_{A1}	0.97, 136.23	0.80, 77.68
P_{A2}	0.93, 80.13	0.69, 60.39
P_{A3}	0.90, 80.63	0.66, 55.87
P_{A4}	0.98, 138.86	0.83, 85.01
P_{A5}	0.98, 136.37	0.83, 83.38
P_{A6}	0.98, 144.04	0.84, 86.90
P_{A7}	0.98, 139.21	0.83, 84.68

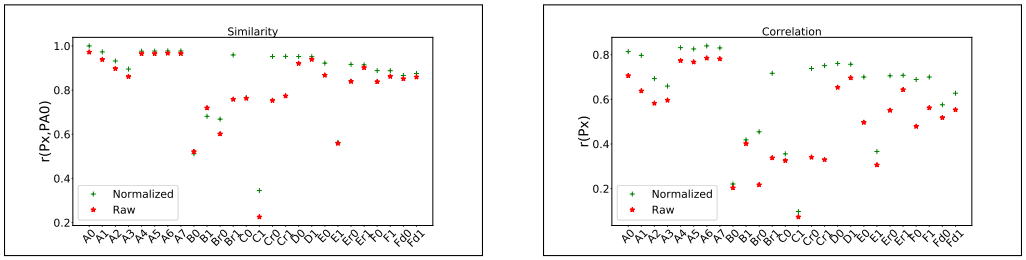


Figure 10: Similarity with the profile of device *A0* (a) and strength of the leak (b) for profiles built on different devices during several acquisition campaigns over a a period of several hours. Apart from the few cases when the quality of a profile is low (due to the variability of the real environment), similarity (a) and correlation (b) are significantly high. The confidence is $-log10(p) > 10$ and $r_z > 10$ even for the worse cases. The suffix r indicates a second acquisition campaign for the cases that had problems. The suffix d indicates the *DAPLink* debugger. The advantage of normalization over raw traces in clearly visible.

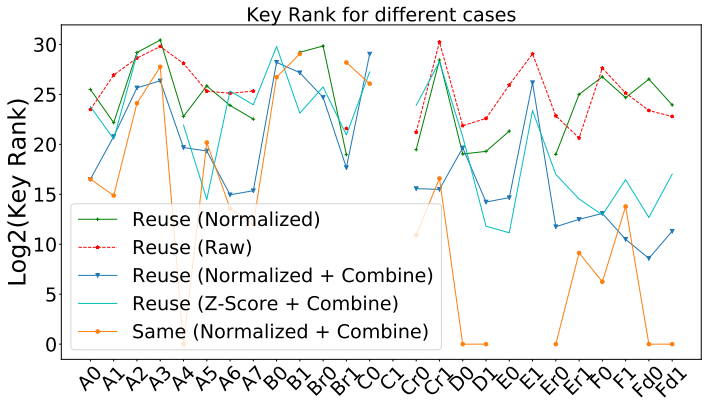
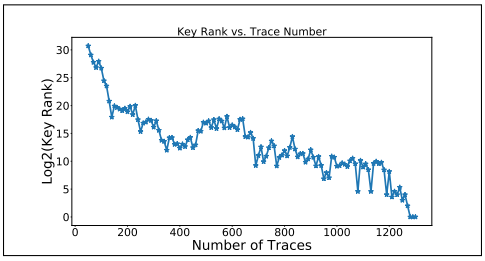


Figure 11: Key rank for different devices and collection campaigns. The baseline (orange) shows the best case (attacking and profiling the same device, normalized traces, average of the 16 bytes as explained in Section 2.6.3). The other lines show the key rank when using the attack traces of device *A0* (first acquisition) with each of the other profiles. In most cases, the key rank shows a moderate increase, which is in the same order of magnitude for time and device changes. Missing points correspond to a key rank bigger than 2^{30}. Our per-trace normalization (blue) has similar results that z-score normalization of the set (cyan).

be faster to bruteforce the remaining bits than to collect more traces. This is particularly true for screaming-channel attacks, because acquisition is slow, and it is only possible when there is an ongoing transmission.

To show a concrete example, we collect 5000·500 template traces and 2000·500 attack traces at a distance of 10 cm in a home environment with a *HackRF* radio on *device E*, at a speed of around 1 trace/s. Figure 12a shows how the key rank decreases slowly with the number of traces. It is therefore faster to collect fewer traces and run key enumeration, provided we do not reach the limit at which enumeration becomes very expensive, as shown in Figure 12b. Using a common laptop[3] with the `AES-NI` instruction, we define three levels of

[3]HP ENVY, Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz, 11GiB Memory.



(a) Key Rank vs. Number of Traces **(b)** Total Attack Time vs. Number of Traces

Figure 12: Analysis of key enumeration, for 5000·500 template traces and 2000·500 attack traces at a distance of 10 cm in a home environment.

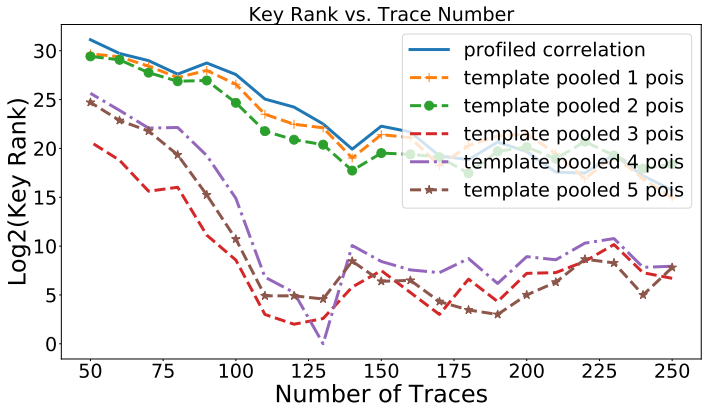


Figure 13: Comparison of multivariate template attacks with profiled correlation, for 5000 · 500 template traces and 2000 · 500 attack traces at a distance of 10 cm in a home environment The attacks with more than one POI clearly converge faster.

enumeration power: quick (up to 2^{20}, a few seconds), medium (up to 2^{30}, a few minutes), slow (up to 2^{37}, several hours). An attacker could parallelize enumeration to improve performance.

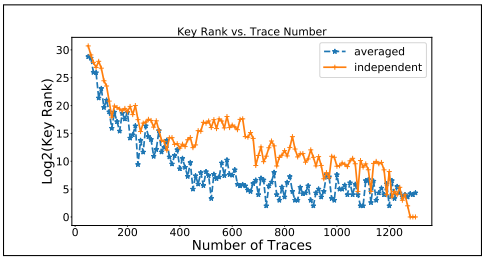
2.6.2 Low Sampling Rate and Informative Points

The clock frequency of the target is 64 MHz. Because of the limitations imposed by the radio hardware and data processing, we sample at a rate of 5 MHz. This limits our ability to capture many informative points in the traces, reducing the interest of dimensionality reduction, multivariate template attacks, and other techniques that combine several POIs.

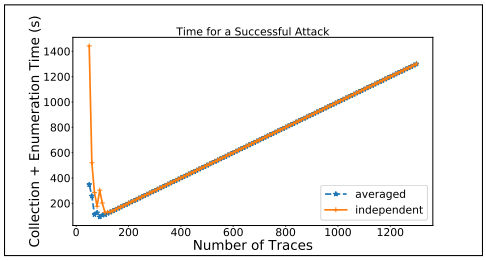
Nevertheless, when we profile in good conditions and with enough traces, we can identify more than one POI, and obtain some advantage from multivariate template attacks. Since collection is slow, and because we have to estimate the leak for 256 values to account for the distortion, the number of traces that we have in each of the 256 classes is normally not large. In this context, using the pooled covariance approach helps to increase the accuracy of the estimate and to avoid computational problems. In the following, we show the advantages of this method in three cases in which it is applicable.

We take the same example as in Section 2.6.1. We run multivariate template attacks with pooled covariance, with variable number of traces and POIs. Figure 13 shows the net advantage over a profiled correlation attack.

We compare a profiled correlation attack and a multivariate attack with pooled covariance



(a) Key Rank vs. Number of Traces



(b) Total Attack Time vs. Number of Traces

Figure 14: Comparison between independent profiles and averaged profiles, for 5000·500 template traces and 2000·500 attack traces at a distance of 10 cm in a home environment.

and 5 POIs on the traces at 10 m in an anechoic chamber. Given 50000·500 template traces, the first attack retrieves the key with $35 \cdot 500$ attack traces and enumeration up to 2^{34}, whereas the second succeeds with 15·500 traces and enumeration up to 2^{31}.

In practice, in our experiments we observe that collection is rarely good enough to extract multiple POIs, and we focus on the use of profiled correlation attacks with one POI. The latter are also computationally more efficient, and thus more useful when the attack set is large (e.g., in challenging scenarios). However, multivariate attacks on higher quality traces remain an interesting direction for future research.

2.6.3 Combining the Profiles of Different Bytes

In a software implementation of *AES-128* such as the `tinyAES` one, there are 16 bytes that leak at 16 different points in time. In our attacks, we profile each of them independently.

For a byte i, given a leak variable $p \oplus k$, each of the 256 possible values of each profile $\hat{model}_{Y_i}(y_i)$ is estimated with $N/256$ measurements, taken at different points in time. Assuming that the profile $\hat{model}_{Y_i}(y_i)$ is the same for each byte i, we can build a single profile using 16 measurements from each trace, instead of 1. Each of the 256 possible values of the single profile $\hat{model}_Y(y)$ is estimated with $16 \cdot N/256$ measurements, taken at different points in time. This brings a gain in accuracy equivalent to collecting 16 times more traces.

The hypothesis that the profile of each byte is the same is reasonable in our case, since the instructions executed for each byte are always the same. We confirm this hypothesis by observing that the correlation between the leakage models of different bytes is very high (in the order of 0.99).

In practice, we still profile the bytes independently, but we add an option to profiled correlation attacks to use a single profile obtained from the average of the 16 different profiles: $\hat{model}_Y(y) = \sum_{i=0}^{15} \hat{model}_{Y_i}(y_i)$.

To show the gain of this combining method, we use the same example as in Section 2.6.1. Results are shown in Figure 14. The advantage of combining the bytes is evident, especially when the number of traces is low. We will show another example in Section 3.4 for an attack at 15 m in an office environment.

This approach and the pooled covariance approach proposed in [CK13] are similar but different, as they operate on different dimensions. We assume that each byte has the same average leak, so that we can compute a pooled estimate of this average leak with a pool of 16 bytes, for each possible value of the leak variable. Instead, the authors of [CK13] assume that the covariance is the same for each value of the leak variable, and they propose to estimate the pooled covariance.

2.6.4 Different Connection Types

In our setup, we use different types of connections between the control laptop, the target, and the radio. In the following, we extend the analysis of our setup to investigate if the type of connection has an impact on the attacks.

The control laptop drives the target (*device E*) through a serial line. The connection is made via a USB cable and a *YKUSH* switchable hub [Yep]. It has individual power drivers for each port, which should reduce coupling and noise propagation from/to the laptop.

The *HackRF* and the *USRP B210* radio are connected to the control laptop through a USB cable. This cable and the one from the YKUSH are connected to the laptop through the same USB hub, upstream w.r.t. the YKUSH to reduce coupling. We have also written a version of the code in which control of the radio and of the target can be split and distributed on two different laptops. In this case the laptops share only the power supply, or nothing if one of the two is powered on battery. Unfortunately, the delays of remote communication between the two laptops make this solution less practical and harder to configure, and this may also have an impact on the quality of the collected traces.

The *USRP N210* uses Ethernet for control and data. We use three different types of network connections with the control laptop. First, a direct connection via an Ethernet cable. Second, a connection that goes through a switch. Third, we connect both the radio and the laptop to two different ports of the Local Area Network (LAN) of the building. Unfortunately, in this last case network delays may cause data loss and reduce the quality of the traces.

At a distance of 10 cm in a home environment, we collect $4700 \cdot 500$ profiling traces for each of the three possible USB connections with a *HackRF* radio, and $4700 \cdot 500$ fixed vs. fixed traces for direct Ethernet connection and connection through the LAN of the building for the *USRP N210*. The maximum correlation found with the ρ-test and its confidence are shown in Table 6. Direct connections lead to better results. It is hard to pin point the exact reason, as indirect connections suffer from larger delays and data losses that may reduce the quality of the traces. We leave a more detailed study of this aspect for future work. For our large distance attacks, we use the *USRP N210* connected via a switch or the LAN of the building. We believe that the ability of the attacker to plug to the same power/Ethernet network as the victim in the same building is a reasonable hypothesis. More comparison data between Ethernet connections will be given in Section 3.4.

3 Challenging Attacks

Thanks to a better understanding of screaming-channel leaks, we can now optimize collection, profiling, and attack. Based on these improvements, we target more challenging scenarios.

3.1 Lessons Learned About Exploitation From the Analysis Phase

From what we have learned about the channel in Section 2, we can draw the following conclusions about exploitation:

- **Collection.** As expected, the radio channel plays and important role. The attacker should maximize the gain to compensate for the quadratic power loss of the leak with distance. The use of a directional antenna and/or of spatial and time diversity

Table 6: Comparison of different connection types.

environment	same USB	same power	floating	direct Ethernet	building LAN
antenna	$p \oplus k$	$p \oplus k$	$p \oplus k$	fixed vs. fixed	fixed vs. fixed
max ρ	0.64	0.12	0.46	0.89	0.77
max r_z	51.64	8.24	33.98	97.99	70.22

helps to reduce several forms of noise, including multi-path reflections and other radio sources. Time diversity is also useful to overcome the deep fade condition that arises when no packet is being transmitted. For GFSK transmissions with this chip, the intended data transmission is otherwise orthogonal to the leak. The quality (e.g., noise figure) of the radio hardware and of the environment (e.g., anechoic chamber) might be more important than the effect of distance. For example, a profile collected in an anechoic chamber with good hardware at 5 m might be better than one taken in a home environment with low-end hardware at 10 cm. Trace collection is particularly slow for screaming channels, because the attacker has to wait for packet transmission to extract useful signals.

- **Profiling.** On our target chip, screaming-channel leaks have a nonlinear leak model, whereas conventional leaks follow the Hamming Weight model. This distortion is mostly independent of distance, setup, and channel frequency. The attacker can capture a nonlinear model by choosing $y = p \oplus k$ and estimating the model for each possible value of y. Thanks to normalization and channel estimation, profiles can be reused in different conditions, at different distances, and against different instances of a device. The attacker can, if necessary, perform profiling in convenient conditions, and attack a different instance in a more challenging setup. The huge advantage on profiling is by far worth the small disadvantage of attacking a different instance. Combining the profiles of different bytes can lead to a better pooled estimate. Given the extremely low sampling speed, the number of POIs is small.

- **Attack.** Attacks have to be able to capture the nonlinearity of the leak model. This is possible thanks to the profiling step of profiled correlation and template attacks (with a good choice of $y = p \oplus k$ to estimate the full profile). We observe that univariate template attacks (which can capture a second order relation between the leak $l(y)$ and the model $model(y)$), do not perform better than profiled correlation attacks (which can only capture the linear correlation between leak and model). We conclude that, for our sample size, looking at linear correlation is enough. Multivariate template attacks are instead more efficient, because the exploit the information of more than one POI. However, given the low sampling frequency and the complex channel, multiple POIs are available only in a few favorable cases (e.g., in the anechoic chamber). Profiled correlation attacks are therefore the preferred attack tool in this paper. They also have the additional advantage of being practical and fast. Key enumeration is often very useful since collection is slow, especially in challenging attack scenarios.

In the following, we will explore more challenging attacks, in more complex and realistic conditions than [CPM+18], and against additional targets. Before continuing, we recall that the best attacks that we reported in [CPM+18] are at 10 m in an anechoic chamber ($70000 \cdot 500$ template traces and $1428 \cdot 500$ attack traces), and at 1 m in an office environment (moving to $52589 \cdot 500$ attack traces). We take the attack at 10 m in an anechoic chamber as reference. To be fair, we apply the improvements that we have learned on the publicly available traces. Results are in Table 7 and show the impact of each technique.

3.2 More Challenging Targets

While non-optimized `tinyAES` is a good benchmark to study screaming-channel effects, we are also interested in evaluating more realistic targets.

3.2.1 Optimized Code

We compile `tinyAES` with higher optimization level (-03), and we investigate the screaming-channel leak at 10 cm with a *HackRF* radio and a standard WiFi antenna. We collect $5000 \cdot 500$

Table 7: Before exploring more challenging attacks, we show as reference the attack at 10 m in an anechoic chamber of Screaming Channels [CPM+18]. To be fair, we apply our improvements (incrementally). We reduce the number of profiling traces and we move to profiled correlation with 1 POI (Less Profiling Traces). Correlation improves significantly when considering the nonlinear leak model (1 Full Profile) and the channel (2 Normalization). Combining the 16 bytes improves the attacks (3 Combining). Univariate template attacks (with pooled variance) are not better (4 Template). Multivariate templates attacks (with pooled covariance) are better when more POIs are available (5 Multivariate).

	Profile	max ρ,r_z	Attack	Type, POIs	Rank
[CPM+18]	130k	0.14, 52	1428	t.a., 10	0
Less Profiling Traces	50k	0.15, 34	3000	p.c., 1	$>2^{33}$
1 Full Profile	50k	0.63, 167	3000	p.c., 1	2^{29}
2 Normalization	50k	0.18, 41	3000	p.c., 1	$>2^{33}$
1 + 2	50k	0.78, 235	3000	p.c., 1	2^{13}
3 Combining + 1 + 2	50k	0.78, 235	1775	p.c., 1	0
3 Combining + 1 + 2	50k	0.78, 235	100	p.c., 1	2^{23}
3 Combining + 1 + 2	50k	0.78, 235	35	p.c., 1	2^{33}
4 Univariate + 1 + 2	50k	0.78, 235	1775	t.a.p.c., 1	2^{18}
4 Univariate + 1 + 2	50k	0.78, 235	100	t.a.p.c., 1	2^{28}
4 Univariate + 1 + 2	50k	0.78, 235	35	t.a.p.c., 1	2^{36}
5 Multivariate + 1 + 2	50k	0.78, 235	1775	t.a.p.c., 5	0
5 Multivariate + 1 + 2	50k	0.78, 235	100	t.a.p.c., 5	2^{23}
5 Multivariate + 1 + 2	50k	0.78, 235	15	t.a.p.c., 5	2^{31}

traces on *device E*. On one side, collection becomes faster, but on the other side the code performs less memory accesses, leading to lower consumption and lower correlation. Figure 15a shows the shape of the leak trace, whereas Figure 15b is the result of the ρ-test. The correlation is around 15 times smaller than for the profile taken in the same conditions on non-optimized code (Section 2.6.1). This shows that an attack is still possible, but with a significantly larger amount of attack traces. For example, for profiled correlation attacks the number of traces required for key recovery increases with the square of the correlation ($N \propto c/\rho^2$) [SPRQ06]. The shape of the leak for the two profiles has a certain degree of similarity, shown in Table 8.

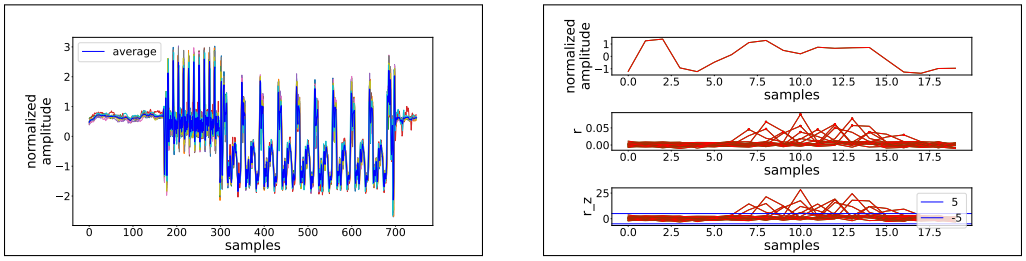
3.2.2 Hardware *AES-128*

The nRF52832 has a dedicated hardware block for *AES-128*, which can be used both in Electronic Cook Book (ECB) and Counter with Cipher Block Chaining Message Authentication Code (CCM) mode. ECB could be useful, for example, to build simple authentication schemes at the application layer [Dan17]. CCM is used by BLE to protect the link layer, and it has priority. It can be configured to encrypt packets concurrently and synchronously to packet transmission, which is a positive point for screaming channels.

Leak Detection. We collect 350000·100 profiling traces on *device F* at 10 cm in an office

Table 8: Correlation among profiles (byte by byte for POI 0) between optimized and non-optimized traces.

byte	0.00	1.00	2.00	3.00	4.00	5.00	6.00	7.00
ρ	0.30	0.32	0.38	0.36	0.20	0.36	0.38	0.18
-log10(p)	6.14	6.86	9.41	8.53	2.84	8.41	9.25	2.50
byte	8.00	9.00	10.00	11.00	12.00	13.00	14.00	15.00
ρ	0.19	0.31	0.50	0.25	0.15	0.24	0.37	0.15
-log10(p)	2.76	6.44	16.94	4.36	1.72	4.10	8.81	1.83



(a) 10·500 traces and their average **(b)** ρ-test

Figure 15: Leak at optimization level O3, at 10 cm in a home environment. The *AES-128* encryption is clearly visible (a). The ρ-test detects the POIs which show significant correlation with the leak variable (b).

environment with a *USRP B210* radio. To simplify collection, we transmit a continuous wave, and we add a preamble (a simple series of nested loops) before calling the encryption function. In our version of the SDK [Nora], the driver copies plaintext and key into a structure read by the hardware block. Similarly, the ciphertext is copied back from this structure to the location required by the user. The leaks originating from p, k, or c, appear on the radio signal, as shown in Figure 16. The hardware block uses a Direct Memory Access (DMA) mechanism to load plaintext and key from RAM memory, and to write back the ciphertext. We are able to observe small correlation peaks for $HW[p]$, $HW[k]$, and $HW[c]$ (Figure 16), due to the hardware block reading and writing data. Using the Hamming Weight is not the best choice to capture distortions, but it requires less traces to observe the peaks, because it reduces the number of classes. Despite the large number of traces and the quality of the setup, we were not able to detect any leak due to the internal operations of the hardware block. Connecting the device via cable and with an additional amplifier did not improve this result.

DPA. Since the internal operations of the hardware block are not visible with the current setup, we were not able to run a DPA attack (e.g., on the output of the Sbox). This result does not exclude the possibility that these leaks are visible with more complex reception hardware, or that the hardware block of other mixed-signal designs is strongly affected by screaming channels.

SPA. The correlation peaks found for the memory transfers of key material used by the hardware block can be attacked using SPA. Besides SPA being harder than DPA, correlation is low, making an attack hard in practice. Despite collecting $1000000 \cdot 100$ attack traces we were not able to reduce the rank of the key enough to run a key enumeration attack. However, attacking the transfer of sensitive material to the hardware block is an interesting open direction for future research.

In practice, we conclude that the hardware cryptographic block of the *nRF52832* chip is as of now by far harder to attack with screaming channels than a software implementation.

3.3 More Challenging Environment

Radio transmissions in real-world environments such as a house are made more complex by the presence of interfering devices and reflections on obstacles. Nevertheless, screaming-channel attacks are still feasible.

3.3.1 0.55 m in Home Environment with Obstacles

In a realistic scenario, it may be hard for an attacker to find a direct line of sight to the target to use directional antennas, but the problem can be solved with spatial diversity.

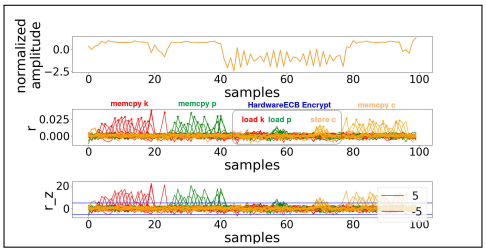


Figure 16: ρ-test on hardware *AES-128*, using $350000 \cdot 100$ traces at 10 cm, in an office environment. Peaks for $HW[k]$ (red), $HP[p]$ (green) and $HW[c]$ (orange) are visible. The bigger peaks correspond to the firmware copying the values to the data structure known by the hardware block. The smaller peaks are due to the hardware block loading those values.

Attack Set for *Device E*. We place our target at 0.55 m from a B210 radio with two receive chains. We use two standard WiFi antennas at a distance of 3 cm from each other. In the middle, we introduce three common off-the-shelf objects with different shapes and materials, which cover the line of sight. Moreover, the table, the curtain, and other walls and obstacles in the room may reflect the signals as well. We collect $2000 \cdot 500$ attack traces in this setup, shown in Figure 20a.

Profiling *Device E* in a Convenient Setup. Before attacking in this challenging condition, we build a profile of the leak at 10 cm in line of sight, using $37000 \cdot 500$ traces, combined from two channels with the maximal ratio technique (same profile as discussed in Section 2.4.3). We then attack using profiled correlation and average of the 16 bytes as explained in Section 2.6.3. We can recover the key using enumeration up to 2^{24}, 2^{28}, and 2^{27}, with $1990 \cdot 500$ traces from channel 1, equal gain, and maximal ratio, respectively. Maximal ratio is better than equal gain, as expected. Channel 1 is much better than channel 2, whose rank is beyond our computational power.

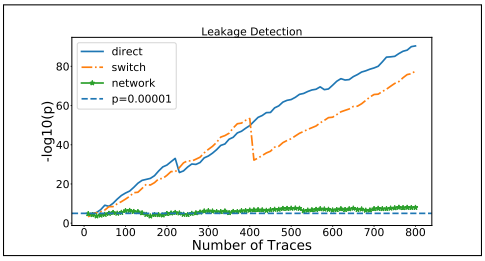
Profiling *Device F* in a Convenient Setup. The attack using maximal ratio succeeds also with a profile ($10000 \cdot 500$ traces) built at 1 m in an office environment with a directional antenna (same setup as in Section 3.4). Since profiling and attack traces were collected with a different configuration for the trigger, we need to manually realign them by applying a fixed offset. The rank increases from 2^{27} to 2^{30}.

3.3.2 1.6 m in Home Environment

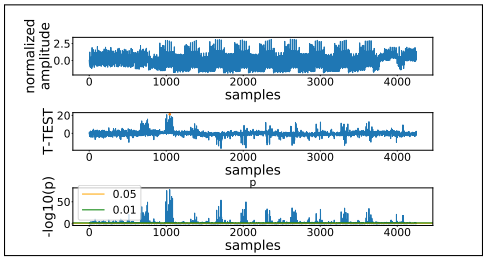
We increase the distance to 1.6 m in a home environment, where the target and the radio are close to walls, tables, and other objects that may reflect. We leverage spatial diversity to improve the results. Figure 20b shows the setup, in which we collected $20000 \cdot 500$ attack traces.

Profiling *Device E* in a Convenient Setup. We use the same profile at 10 cm as Section 2.4.3 and Section 3.3.1. Using the traces combined from the two channels with the maximal ratio technique, a profiled correlation attack succeeds after enumeration up to 2^{31}. With the traces combined with equal gain, enumeration succeeds at 2^{35}, whereas channel 1 requires 2^{31} and channel 2 needs more than 2^{34} (then we time out).

Profiling *Device G* in a Convenient Setup. The attack using maximal ratio succeeds also with a profile ($10000 \cdot 500$ traces) built on another device connected via cable (see Section 3.4). Since profiling and attack traces were collected with a different configuration for the trigger, we need to manually realign them by applying a fixed offset. The rank increases from 2^{31} to 2^{34}.



(a) Leakage Detection at 10 m



(b) t-test at 10 m

Figure 17: At 10 m we can easily detect the leak with all types of connections (a). The t-test has peaks for all data-dependent points, including the output of the Sbox, which can be used to attack (b).

3.4 More Challenging Distance

One of the peculiar features of screaming-channel leaks is that they can be observed at large distance. We study from how far we can attack in an office environment. A simplified scheme of the setup is shown in Figure 20c.

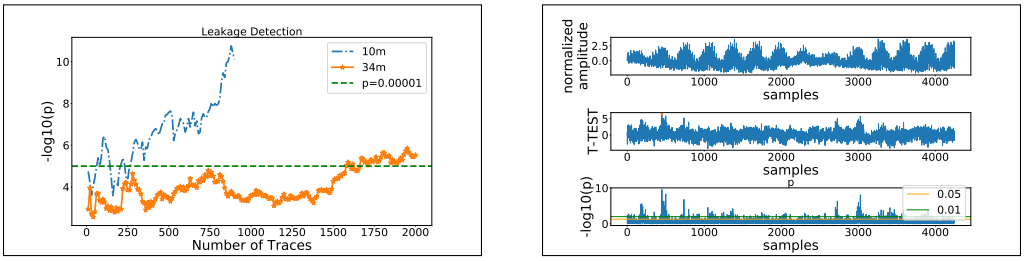
3.4.1 10 m in Office Environment

In an office environment, we set target device and radio receiver 10 m apart. To avoid reflections, and to increase the gain, we use a directional antenna with 24 dBi gain ([TP-]), and two low-noise amplifiers with 20 dB gain and 1.5 dB noise figure ([Mina]). This is the same setup used in Screaming Channels for the anechoic chamber. We additionally test three different connections with the *USRP N210* radio: direct Ethernet cable, connection through a switch, and connection through the Ethernet network of the building. In the last case we experience packet loss due to the speed of the network, but we are still able to collect traces.

Leakage Detection. As a first step, we evaluate the presence of the leak with a t-test. Following [DS16], we choose a fixed vs. fixed configuration in order to maximize the leak signal, and consequently reduce the number of traces required for detection. However, because of the distortion seen in Section 2, the maximum difference appears for $p \oplus k = 0$ and $p \oplus k = 48$, instead of $HW[p \oplus k] = 0$ and $HW[p \oplus k] = 8$. Therefore, we collect two sets $\mathcal{L}_{p \oplus k=0}$ and $\mathcal{L}_{p \oplus k=48}$ of $400 \cdot 500$ traces each. To be precise, we run 1000 encryptions and select the first 500 of them. This ensures we always average 500 traces despite sudden sources of noise, network delays, holes between packets, and other reasons for which we might miss the extraction of some encryptions. The number 500 tells us about the actual number of traces we need at 10 m, whereas the number 1000 give us a rough idea of how much harder it is to extract traces in a more complex environment or setup. We consider that the t-test successfully detects a leak when the maximum value $max(|t|)$ has $p < 10^{-5}$. With a few tens of traces the leak appears for each type of connection, as shown in Figure 17a. Detection is slower for the connection through the network of the building, most likely because of data loss. Figure 17b shows that the leak at the output of the first Sbox is clearly visible.

Attack Set for *Device F* at 10 m. Once we have confirmed that the leak is visible even at 10 m in an office environment, we collect an attack set for *device F*. We acquire $6000 \cdot 500$ traces (actually 1000 encryptions for each trace, as for detection), connecting to the radio through a switch. Collecting a profile set in this scenario would be complex and it would require a large amount of traces to overcome noise. Fortunately, as seen in Section 2.4 and Section 2.5, profiles can be reused across distances, setups, and against different devices. Therefore, we can collect the profile set in more convenient conditions, and reuse it.

Profiling *Device G* in a Convenient Setup. We collect a profile set on another



(a) t-test at 10 m and 34 m

(b) t-test at 34 m, 4000·500 traces

Figure 18: The t-test succeeds even at 34 m.

instance (*device G*), which we have modified for direct connection via cable with a *HackRF* radio. The main advantage of the cable connection is that we can pay much less attention to the noise in the environment. Moreover, using a simpler radio and no external amplifier greatly reduces costs and complexity. In the end, a laptop with two USB ports is enough to collect 10000·500 traces for the profile in a few hours.

Successful Attack. With a profiled correlation attack (leak variable $p \oplus k$, average of the profiles of the 16 bytes, one POI) and key enumeration up to 2^{28} we can recover the key using 1500·500 attack traces at 10 m.

3.4.2 15 m in Office Environment

We increase the distance to 15 m, using the same setup and configuration as at 10 m.

Attack Set for *Device A* at 15 m. We collect $7000 \cdot 500$ traces (actually 1000 encryptions for each trace, as before). We observe that a meticulous tuning of the setup is particularly important in these conditions. We need to calibrate the *USRP N210* radio right before the acquisition campaign, and to position the directional antenna with care. It took us several attempts before managing to collect a set of traces good enough for an attack. On the contrary, at shorter distances screaming channels are easy to mount with simpler and cheaper hardware, without need of extremely accurate tuning. We conclude that, as expected, screaming-channel attacks can be easier or harder to mount than conventional attacks, depending on distance and environment.

Profiling *Device G* in a Convenient Setup. As before, we assume that the attacker can profile a different device with a convenient connection by cable, using 10000·500 traces. These profiling traces were collected 5 months and 15 days before the attack traces. To sum up, profiling and attack traces differ in device instance, propagation medium and distance, antenna and receiver type and quality, and acquisition time.

Successful Attack. Using profiled correlation (leak variable $p \oplus k$, average of the 16 bytes, one POI), the attack succeeds with 5000·500 traces and key enumeration up to 2^{23}.

3.4.3 34 m in Office Environment

After attacking at 15 m, we increase the distance to 34 m, keeping the same setup and configuration. In this case we connect the laptop and the *USRP N210* through the Ethernet network of the building (at the price of some packet loss). We conduct the same fixed vs. fixed t-test experiment as we did at 10 m. The leak is still visible and the t-test succeeds, as shown in Figure 18.

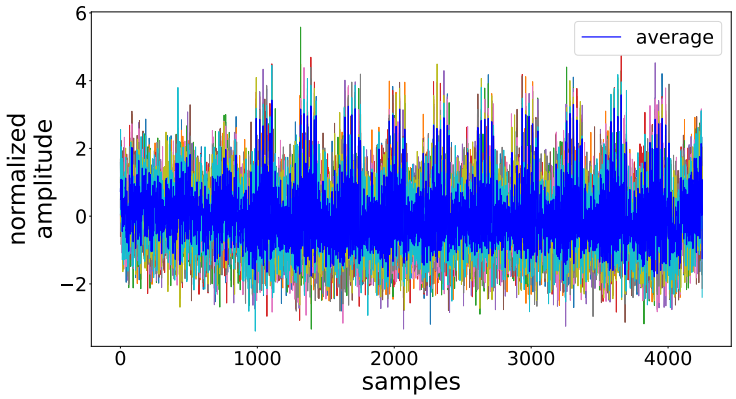


Figure 19: 10·500 traces and their average, at 60 m in an office environment. The *AES-128* encryption is still clearly visible.

3.4.4 60 m in Office Environment

The detection of leak traces in the radio signal is based on a band pass filter, tuned into a characteristic frequency component of the leak trace. Since this component has a very narrow band, we can use a very narrow filter. For example, for the setup at 10 m we use a 6^{th}-order Butterworth filter with cutoff frequencies of 1.95 MHz and 2.02 MHz. The noise in such a small bandwidth will be rather small. As a consequence, trace detection is quite robust to noise, and we expect to be able to extract traces even at distances larger than 34 m. Keeping the same setup, we place the target at 60 m from the antenna. Figure 19 shows 10·500 traces and their average. The key schedule and the 10 rounds of *AES-128* are clearly visible.

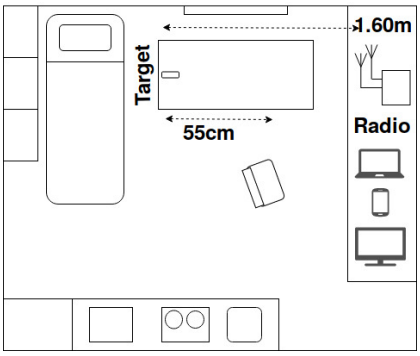
3.5 Summary of Results

In this section we have leveraged our analysis and attack improvements in order to investigate more challenging scenarios than Screaming Channels. First, we have shown the level of correlation for optimized code and hardware *AES-128*, which gives an idea of the complexity of an attack. Second, we have used spatial diversity to attack in a real home environment, with obstacles and other sources of reflections. Finally, we have mounted a full attack at 10 m and 15 m in a real office environment, by combining the profiles of multiple bytes and with efficient key enumeration. We were able to reuse profiles built in more convenient setups on a different device than the one under attack. We could detect the leak at 34 m with a fixed vs. fixed t-test, and extract traces at 60 m, still in a real environment. These are all significant steps towards realistic exploitation of the screaming-channel vector.

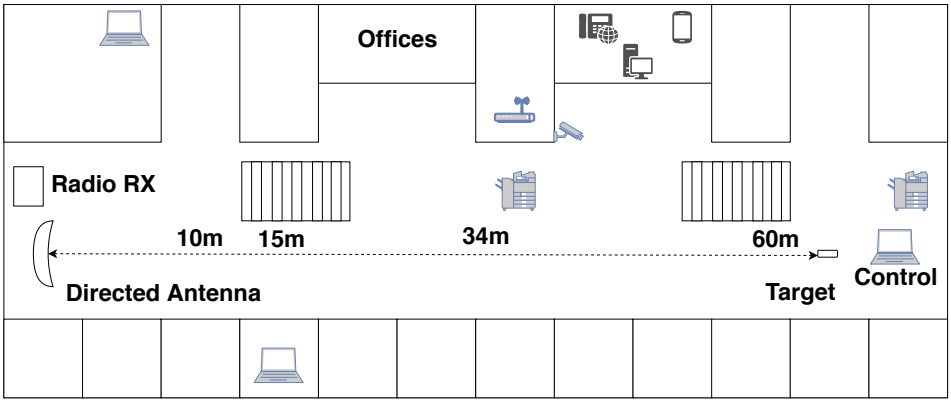
4 Proof-of-concept Attack Against a Real System

The purpose of this paper is to uncover the properties of an unknown channel, and to mount successful attacks in more challenging and realistic conditions than previous work. Our results could be the basis for a preliminary analysis of the risks associated with screaming channels in real systems and protocols. To provide a concrete example, we study the case of authentication for Google Eddystone beacons, and we show a proof-of-concept attack.

(a) Obstacles in a home environment (0.55 m)



(b) Home environment (0.55 m, 1.60 m)



(c) Office environment (10 m, 15 m, 34 m, 60 m)

Figure 20: Some more challenging attack setups.

4.1 Google Eddystone Beacons

BLE beacons are devices designed to be placed at strategic places and continuously transmit some information on the advertising channels. People passing nearby can receive such messages on their mobile phones. Beacons can be used for numerous applications, ranging from marketing in shopping malls to indoor location services. For example, they can be placed near a monument and broadcast the URL of a webpage with information about it. There exist several beacon formats, such as, Apple iBeacons, Radius Networks AltBeacons, and Google Eddystone.

Eddystone [Gooa] is a beacon format introduced by Google in 2015. Its design includes some interesting security features that ensure confidentiality and integrity of telemetry data, protect from spoofing and tracking, and require authentication to access and configure the beacon. One of their uses is to build the Physical Web [Goob], where objects broadcast a URL. There are four frame types:

- **Eddystone-UID.** The identifier of the beacon.
- **Eddystone-URL.** The frame used to broadcast a URL.
- **Eddystone-(e)TML.** The frame used to broadcast (encrypted) telemetry data.
- **Eddystone-EID.** An ephemeral identifier that can be resolved only by a registered service, whereas it looks random to others.

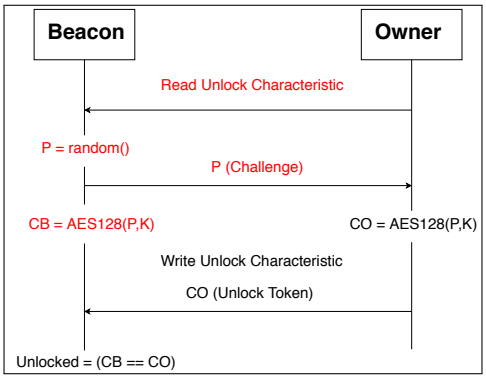


Figure 21: Unlock scheme used by Google Eddystone Beacons. Note that a non-authenticated device can trigger encryptions with known plaintext by reading the unlock characteristic (steps marked in red).

To configure a beacon (e.g., to register an EID), the owner has to connect to it using another BLE device that acts as master, and use the Eddystone Configuration GATT Service. Some beacons go in the connectable state after some kind of input (e.g., a button), other beacons have to be always connectable (e.g., if they are not easily accessible). Before being able to interact with the beacon, the owner has to unlock it using a 128-bit key (unlock code). The unlock method is shown in Figure 21. The device that wants to authenticate reads the unlock characteristic of the beacon. The beacon generates a random challenge that it sends back. Then it creates an unlock token by encrypting the random challenge with a preshared unlock code. Similarly, the owner creates the unlock code on the other side, then it writes it in the unlock characteristic. Finally, the beacon checks if the unlock code computed by the device that wants to authenticate is correct, and if so it opens the lock. The advantage of this method is that the unlock code is never sent in plaintext over the air. An attacker can read the unlock code, but it is a useless information because a new challenge is generated at each new unlock request. The disadvantage is that a device that connects to the beacon can trigger encryptions with known plaintext, which is a good scenario for side-channel attacks.

4.2 Proof-of-concept Attack Against Google Eddystone Authentication

We observe that the Google Eddystone unlock method is a good target for screaming-channel attacks.

4.2.1 Threat Model

Let us imagine the following scenario. A fleet of beacons is located in a position which is not easily accessible. For this reason, and to easily configure many beacons without manual access, they are configured to always stay in connectable mode. To avoid malicious reconfiguration (e.g., broadcast of a malicious URL), they are protected with a lock. The unlock scheme lets an attacker force many encryptions with known plaintexts, which is a favorable scenario for side-channel attacks. However, to exploit a conventional side channel, an attacker would have to get physical access to the devices, which are in a non easily accessible place, maybe even in a protected area such as a shopping mall. For these reasons, side-channel attacks are not taken into account in the threat model, and the unlock encryption is not protected. A countermeasure against side channels would also increase the cost of the device. As also highlighted in Screaming Channels, this is a reasonable assumption for this kind of devices.

While conventional side channels require physical access, a screaming-channel attack could be performed discretely from a certain distance. This is a new threat to take into account.

4.2.2 Real Target with Optimizations and Frequency Hopping Enabled.

For this proof-of-concept attack, we target the Google Eddystone demo available in the manufacturer's SDK [Nora]. In this version, the encryption is performed by default with the `tinyAES` library compiled with optimization level O3[4]. We take the demo as is, without any modification, and we flash it on a *PCA10040* board. This is a realistic target with optimized code, frequency hopping enabled, and higher layers and protocols built on top of BLE.

4.2.3 Trace Collection

Using a mobile phone app, we simulate the owner of the device and we configure it to be always connectable, then we lock it. We can also set the transmission and advertising power to 4 dBm. We extended the collection code to be able to interact with any Eddystone beacon using the BLE card of the computer or an external off-the-shelf BLE dongle. The code can set a new random key, and then lock the device. It can also force many encryptions with known plaintext by repeatedly reading the unlock characteristic and saving the answer.

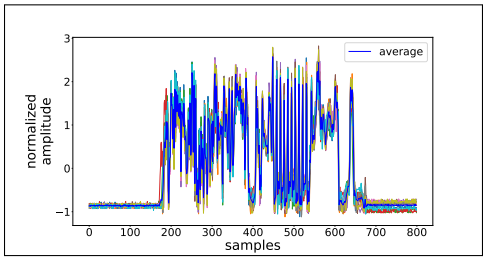
Minimizing the Frequency Hopping Problem. In the scenario we consider, the attacker connects to the target device as a BLE master. Therefore, we can minimize the problem of frequency hopping by reducing the number of channels actually used for communication, as seen in Section 2.2.3. Following the BLE [SIG16] specifications, we craft a *Set Host Channel Classification Command* to blacklist all data channels but 0,1,16,34. On the Linux host of the attacker, we issue this command to the BLE dongle that acts as master using the *hcitool* program, as shown in Listing 1. Once set, the new channel map will be valid for the following connections as well.

Listing 1: Minimizing Frequency Hopping.

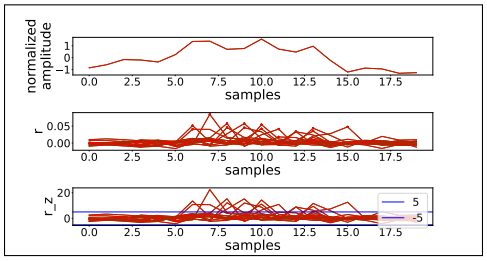
```
> # Minimize Frequency Hopping
> sudo hcitool lecc --random E5:8F:A5:90:43:80 # connect
> sudo hcitool cmd 0x08 0x0014 0x0000000003 # set channel map
> sudo hcitool ledc 64 # disconnect
>
> # Double Check
> sudo btmon # monitor hci (in another terminal)
>
> sudo hcitool lecc --random E5:8F:A5:90:43:80 # connect
> sudo hcitool cmd 0x08 0x0015 0x40 # read channel map
>
> # Excerpt of the Expected Output (btmon):
> # LE Read Channel Map (0x08|0x0015) ncmd 1
> #   Status: Success (0x00)
> #   Handle: 64
> #   Channel map: 0x0300010004
> #     Channel 0-1
> #     Channel 16
> #     Channel 34
>
> sudo hcitool ledc 64 # disconnect
```

Extracting Encryption Traces. We start triggering encryptions in the beacon by reading the unlock characteristic. To have an idea of the leak trace to expect, we first extract

[4] More recent versions of the SDK allow choosing other options such as hardware encryption, which is the default. However, we still think that software encryption with `tinyAES` is a reasonable and interesting example which might be found in the wild. In general, application code might prefer software implementations for portability, or to avoid competing with the link layer for the use of the hardware encryption block.



(a) Key schedule and first rounds for Eddystone



(b) ρ-teset for Eddystone

Figure 22: The key schedule and the first rounds of *AES-128* are clearly visible in the traces we collect for Eddystone (a). The correlation peaks for $p \oplus k$ are clearly present as well (b).

the encryption trace with a conventional setup at 64 MHz. Here we observe a trace that is similar to Figure 15a, which is around 46 µs long. We then move to a screaming-channel setting. To minimize interference from other transmissions in the ISM band, we tune at the second harmonic of the clock after channel 34: $f = 2.474\text{GHz} + 2 \cdot 64\text{MHz} = 2.602\text{GHz}$. By reading the firmware, we observe that the encryption happens quickly after the read request to the unlock characteristic and the generation of the random unlock challenge. The response to the other device (containing the random challenge) is sent only after the encryption is complete, therefore we cannot expect that packet to contain any useful trace. We have to rely on other packets transmitted at the same time of the encryption. We notice that the device always sends a packet just after receiving the unlock characteristic read request. This might be the acknowledgement packet generated at the link layer. The encryption and this packet are almost synchronous. Most of the time the key schedule is visible in the leak towards the end of the packet, as shown in Figure 22a. On a certain number of packets the first *AES-128* round is visible as well. We extract this signal containing the key schedule and the beginning of the first round, and we use it as template to collect more. Every time we trigger one encryption, we extract the leak corresponding to the packet, and then we correlate it with the template. If correlation is high enough to confirm that we found the desired signal, we align it and save it. Note that in this case we cannot use time diversity, and we collect single encryptions.

4.2.4 Key Recovery Attack

Once we have confirmed that we can extract encryption traces with known plaintext, we move to the attack phase.

Profiling *Device H* in a Convenient Setup. For simplicity, we connect to *device H* via cable, and we collect $70000 \cdot 1$ profiling traces with random plaintext and a fixed random key, over a period of around 3 days. Correlation peaks for $y = p \oplus k$ are clearly visible in Figure 22b.

Attacking *Device H* in a Convenient Setup. We collect $33000 \cdot 1$ traces. We run a profiled correlation attack with leak variable $p \oplus k$. To obtain a better result, we replace the trace point corresponding to a POI with the average of the three points around the POI. This is a simple way to capture the information present just around the POI, without resorting to more complex multivariate attacks. We can recover the key with key enumeration up to 2^{30}.

4.3 Countermeasures

The simplest countermeasure to this attack consists in limiting the number of reads to the unlock characteristic that are not followed by a successful write of the unlock code. Alternatively, the answer to the read could be made slower after a certain number of attempts. The beacon could move from using a software implementation of *AES-128* to using the

hardware block[5]. Indeed, we have seen that it appears harder to exploit due to its lower power consumption. However, using the hardware implementation would tie the code of the application to a particular hardware peripheral, making it less generic and portable. If the use case allows it, the beacon could be configured to enter the connectable state only if the (legitimate) users press a button. However, in our threat model we consider the case in which the device is not easily accessible and has to stay in connectable mode. Finally, screaming channels could be countered with conventional protections such as masking or addition of noise. However, the (low) price of these devices might not allow for the licensing or development costs required for these additions.

5 Conclusions

Screaming channels are an interesting leak vector, with peculiar characteristics compared to other forms of side channels.

The fact that analog leaks propagate over an intended digital radio channel is at the same time a tremendous advantage and a new problem. On the one hand, the leaks profit from dedicated transmission hardware, making side-channel attacks possible at large distance, as highlighted in the seminal Screaming Channels paper. On the other hand, the analog leaks have to coexist with the intended data sent as digitally modulated discrete packets, making reception more complex. Sometimes, such problems can be addressed at the system level, for example, minimizing frequency hopping by blacklisting most channels. While the amount of correlation of screaming channels is comparable to conventional leaks, the leak model is nonlinearly distorted. This forbids the use of simple leak models, and requires the full profiling of the leak function. Fortunately, with proper normalization, profiles can be reused without considerable loss across distances, setups, acquisition campaigns, and device instances. The reception setup is different from that of conventional attacks. At short distances it can be simpler and cheaper. As distance increases, the quality of the setup, the type of antenna, and spatial diversity, start playing a fundamental role. Proof-of-concept attacks succeed at 15 m in an office environment. At larger distances the leak is visible, but too challenging to exploit with the current hardware setup and signal processing. As of now, attacks are not possible on the cryptographic hardware block. However, the memory transfers of key/plaintext/ciphertext bytes to the hardware are visible, opening an interesting research direction for attacks. A successful attack against the authentication method of Google Eddystone beacons without modifications shows that screaming channels are a real threat to real systems and protocols.

On the one hand, this paper shows the importance of the novel channel through a systematic analysis and several attacks. On the other hand, the analysis and the quantitative results from the attacks help defenders and radio engineers to better evaluate risks and countermeasures.

6 Replicability

To ensure that our result are easily replicable, we provide detailed instructions, code, and pre-collected data at https://github.com/eurecom-s3/screaming_channels.

Acknowledgments

The authors acknowledge the support of SeCiF project within the French-German Academy for the Industry of the future as well as the support by the DAPCODS/IOTics ANR 2016 project (ANR-16-CE25-0015).

[5] This is the default choice in the newest versions of the SDK.

References

[AARR02] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM side-channel(s). In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 29–45. Springer, 2002.

[ant] ANT Wireless Website. https://www.thisisant.com/.

[ARR03] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel attacks. In Colin D. Walter, Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 2–16. Springer, 2003.

[BCD+13] George Becker, J Cooper, Elke DeMulder, Gilbert Goodwill, Joshua Jaffe, G Kenworthy, T Kouzminov, A Leiserson, M Marson, Pankaj Rohatgi, et al. Test vector leakage assessment (TVLA) methodology in practice. In *International Cryptographic Module Conference*, volume 1001, page 13, 2013.

[BJh15] Liu Biao and Kong Jun-hui. Practical template attacks based on pooled covariance matrix. In *2015 7th Asia-Pacific Conference on Environmental Electromagnetics (CEEM)*, pages 184–188, 11 2015.

[BKM+15] Andrey Bogdanov, Ilya Kizhvatov, Kamran Manzoor, Elmar Tischhauser, and Marc Witteman. Fast and memory-efficient key recovery in side-channel attacks. In Orr Dunkelman and Liam Keliher, editors, *Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers*, volume 9566 of *Lecture Notes in Computer Science*, pages 310–327. Springer, 2015.

[BLvV15] Daniel J. Bernstein, Tanja Lange, and Christine van Vredendaal. Tighter, faster, simpler side-channel security evaluations beyond computing power. *IACR Cryptology ePrint Archive*, 2015:221, 2015.

[Bre59] Donald G. Brennan. Linear diversity combining techniques. *Proceedings of the IRE*, 47(6):1075–1102, 1959.

[CCC+19] Mathieu Carbone, Vincent Conin, Marie-Angela Cornelie, François Dassance, Guillaume Dufresne, Cécile Dumas, Emmanuel Prouff, and Alexandre Venelli. Deep learning to evaluate secure RSA implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):132–161, 2019.

[CEK18] Emmanuel Cottais, José Lopes Esteves, and Chaouki Kasmi. Second order soft-TEMPEST in RF front-ends: Design and detection of polyglot modulations. *2018 International Symposium on Electromagnetic Compatibility (EMC EUROPE)*, pages 166–171, 2018.

[CK13] Omar Choudary and Markus G. Kuhn. Efficient template attacks. In Aurélien Francillon and Pankaj Rohatgi, editors, *Smart Card Research and Advanced Applications - 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers*, volume 8419 of *Lecture Notes in Computer Science*, pages 253–270. Springer, 2013.

[CK14] Omar Choudary and Markus G. Kuhn. Template attacks on different devices. In Emmanuel Prouff, editor, *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, volume 8622 of *Lecture Notes in Computer Science*, pages 179–198. Springer, 2014.

[CK18] Marios O. Choudary and Markus G. Kuhn. Efficient, portable template attacks. *IEEE Trans. Information Forensics and Security*, 13(2):490–501, 2018.

[CPM+18] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. Screaming channels: When electromagnetic side channels meet radio transceivers. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 163–177. ACM, 2018.

[CRR02] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

[Dan17] Daniel Veilleux. Simple Application-level Authentication, 2017. https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/simple-application-level-authentication.

[DFS19] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version. *J. Cryptology*, 32(4):1263–1297, 2019.

[DGD+19] Debayan Das, Anupam Golder, Josef Danial, Santosh Ghosh, Arijit Raychowdhury, and Shreyas Sen. X-DeepSCA: Cross-device deep learning side channel attack. In *Proceedings of the 56th Annual Design Automation Conference 2019, DAC 2019, Las Vegas, NV, USA, June 02-06, 2019*, page 134. ACM, 2019.

[DS16] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 240–262. Springer, 2016.

[ECK19] José Lopes Esteves, Emmanuel Cottais, and Chaouki Kasmi. Second order soft Tempest: From internal cascaded electromagnetic interactions to long haul covert channels. *2019 URSI Asia-Pacific Radio Science Conference (AP-RASC)*, pages 1–3, 2019.

[EG12] M. Abdelaziz Elaabid and Sylvain Guilley. Portability of templates. *J. Cryptographic Engineering*, 2(1):63–74, 2012.

[emp] Construction guide for EM probe. https://github.com/emsec/SCATools/wiki/BuildEMProbe.

[Etta] Ettus Research. USRP N210. http://www.ettus.com/all-products/un210-kit/.

[Ettb] Ettus Research. USRP V210. http://www.ettus.com/all-products/UB210-KIT/.

[GBTP08] Benedikt Gierlichs, Lejla Batina, Pim Tuyls, and Bart Preneel. Mutual information analysis. In Elisabeth Oswald and Pankaj Rohatgi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2008.

[GDD+19] Anupam Golder, Debayan Das, Josef Danial, Santosh Ghosh, Shreyas Sen, and Arijit Raychowdhury. Practical approaches toward deep-learning-based cross-device power side-channel attack. *IEEE Trans. Very Large Scale Integr. Syst.*, 27(12):2720–2733, 2019.

[GGJR+11] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[GGP+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 Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 117–129. Springer, 2015.

[Gooa] Google. Eddystone. https://github.com/google/eddystone.

[Goob] Google. Physical web. https://google.github.io/physical-web/.

[Gre17] Great Scott Gadgets. HackRF one, 2017. https://greatscottgadgets.com/hackrf/.

[Gro18] EURECOM S3 Group. Screaming Channels open-source code, 2018. https://github.com/eurecom-s3/screaming_channels.

[GZ13] Nina Golyandina and Anatoly Zhigljavsky. *Singular Spectrum Analysis for Time Series*. Springer Science & Business Media, 01 2013.

[HOTM14] Neil Hanley, Máire O'Neill, Michael Tunstall, and William P. Marnane. Empirical evaluation of multi-device profiling side-channel attacks. In *2014 IEEE Workshop on Signal Processing Systems, SiPS 2014, Belfast, United Kingdom, October 20-22, 2014*, pages 226–231. IEEE, 2014.

[LPR13] Victor Lomné, Emmanuel Prouff, and Thomas Roche. Behind the scene of side channel attacks. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 506–525. Springer, 2013.

[MBTL13] David P. Montminy, Rusty O. Baldwin, Michael A. Temple, and Eric D. Laspe. Improving cross-device attacks using zero-mean unit-variance normalization. *J. Cryptographic Engineering*, 3(2):99–110, 2013.

[MGDS10] Olivier Meynard, Sylvain Guilley, Jean-Luc Danger, and Laurent Sauvage. Far correlation-based EMA with a precharacterized leakage model. In Giovanni De Micheli, Bashir M. Al-Hashimi, Wolfgang Müller, and Enrico Macii, editors, *Design, Automation and Test in Europe, DATE 2010, Dresden, Germany, March 8-12, 2010*, pages 977–980. IEEE Computer Society, 2010.

[Mina] Minicircuits. ZEL-1724LN. https://www.minicircuits.com/pdfs/ZEL-1724LN+.pdf.

[Minb] Minicircuits. ZKL-1R5. https://www.minicircuits.com/pdfs/ZKL-1R5+.pdf.

[Minc] Minicircuits. ZX60-272LN. https://ww2.minicircuits.com/pdfs/ZX60-272LN+.pdf.

[MOBW13] Luke Mather, Elisabeth Oswald, Joe Bandenburg, and Marcin Wójcik. Does my device leak information? An a priori statistical power analysis of leakage detection tests. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 486–505. Springer, 2013.

[MOOS15] Daniel P. Martin, Jonathan F. O'Connell, Elisabeth Oswald, and Martijn Stam. Counting keys in parallel after a side channel attack. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 313–337. Springer, 2015.

[MRSS18] Amir Moradi, Bastian Richter, Tobias Schneider, and François-Xavier Standaert. Leakage detection with the x2-test. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):209–237, 2018.

[Nora] Nordic Semiconductor. nRF5. https://developer.nordicsemi.com/nRF5_SDK/nRF5_SDK_v14.x.x/nRF5_SDK_14.2.0_17b948a.zip.

[Norb] Nordic Semiconductor. PCA10040 Development Kit. https://www.nordicsemi.com/Software-and-tools/Development-Kits/nRF52-DK.

[NSA82] NSA. NACSIM 5000, Tempest fundamentals. Technical report, NSA, 1982. Document declassified in 2000 and available at https://cryptome.org/jya/nacsim-5000/nacsim-5000.htm.

[OM07] Elisabeth Oswald and Stefan Mangard. Template attacks on masking - resistance is futile. In Masayuki Abe, editor, *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings*, volume 4377 of *Lecture Notes in Computer Science*, pages 243–256. Springer, 2007.

[OP11] David Oswald and Christof Paar. Breaking Mifare DESFire MF3ICD40: power analysis and templates in the real world. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2011.

[Par] Particle. Particle Debugger. https://store.particle.io/products/particle-debugger.

[PS15] Santos Merino Del Pozo and François-Xavier Standaert. Blind source separation from single measurements using singular spectrum analysis. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 42–59. Springer, 2015.

[PSG16] Romain Poussier, François-Xavier Standaert, and Vincent Grosso. Simple key enumeration (and rank estimation) using histograms: An integrated approach. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*, volume 9813 of *Lecture Notes in Computer Science*, pages 61–81. Springer, 2016.

[Red] Redbear. BLE Nano v2. https://fccid.io/2AKGS-MBN2.

[RSV+11] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.

[Sema] Nordic Semiconductor. Electronic ShockBurst. https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.sdk5.v15.0.0/group__nrf__esb.html.

[Semb] Nordic Semiconductor. Gazel. https://infocenter.nordicsemi.com/topic/com.nordic.infocenter.sdk5.v15.0.0/gzll_02_user_guide.html.

[SIG16] Bluetooth SIG. Bluetooth 5.0 Core Specification, 2016. https://www.bluetooth.com/specifications/archived-specifications/.

[SKS09] François-Xavier Standaert, François Koeune, and Werner Schindler. How to compare profiled side-channel attacks? In Michel Abdalla, David Pointcheval, Pierre-Alain Fouque, and Damien Vergnaud, editors, *Applied Cryptography and Network Security, 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2-5, 2009. Proceedings*, volume 5536 of *Lecture Notes in Computer Science*, pages 485–498, 2009.

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In Josyula R. Rao and Berk Sunar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.

[SPRQ06] François-Xavier Standaert, Eric Peeters, Gaël Rouvroy, and Jean-Jacques Quisquater. An overview of power analysis attacks against field programmable gate arrays. *Proceedings of the IEEE*, 94(2):383–394, 2006.

[TP-] TP-LINK. TL-ANT2424B. https://www.tp-link.com/us/business-networking/antenna-and-accessory/tl-ant2424b/.

[VGRS12] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In Lars R. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pages 390–406. Springer, 2012.

[VGS13] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Security evaluations beyond computing power. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, volume 7881 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2013.

[vWWB11] Jasper G. J. van Woudenberg, Marc F. Witteman, and Bram Bakker. Improving differential power analysis by elastic alignment. In Aggelos Kiayias, editor, *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, volume 6558 of *Lecture Notes in Computer Science*, pages 104–119. Springer, 2011.

[VYG92] Robert Vautard, Pascal Yiou, and Michael Ghil. Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena*, 58(1-4):95–126, 1992.

[Wel47] Bernard L. Welch. The generalization of 'student's' problem when several different population variances are involved. *Biometrika*, 34(1-2):28–35, 01 1947.

[WO15] Carolyn Whitnall and Elisabeth Oswald. Robust profiling for dpa-style attacks. In Tim Güneysu and Helena Handschuh, editors, *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, volume 9293 of *Lecture Notes in Computer Science*, pages 3–21. Springer, 2015.

[YEM14] Xin Ye, Thomas Eisenbarth, and William Martin. Bounded, yet sufficient? How to determine whether limited side channel information enables key recovery. In Marc Joye and Amir Moradi, editors, *Smart Card Research and Advanced Applications - 13th International Conference, CARDIS 2014, Paris, France, November 5-7, 2014. Revised Selected Papers*, volume 8968 of *Lecture Notes in Computer Science*, pages 215–232. Springer, 2014.

[Yep] Yepkit. YKUSH. https://www.yepkit.com/products/ykush.