Exploring Leakage Characteristics and Attacks through Profiles of Screaming Channels

Yuki Matsukawa, Daiki Miyahara, Takeshi Sugawara, Kazuo Sakiyama, and Yang Li

The University of Electro-Communications, Tokyo, Japan
{y.matsukawa, miyahara, sugawara, sakiyama, liyang}@uec.ac.jp

Abstract

Recent advancements have introduced screaming channels, a novel side channel where information leakages can propagate over extended distances. These developments have considerably threatened the design and security of particular mixed-signal chips. However, conventional research methods require extensive profiling prior to launching attacks, as the Hamming weight model is unsuitable for a screaming channel attack. Through the profiling conducted on BLE Nano V2, similar to previous studies, we observed that in tinyAES, the upper 4 bits of the S-box input exhibit leakage characteristics via the screaming channels. Based on these observations, we confirmed it is possible to recover keys for tinyAES by making assumptions that differ from simple Hamming weights, without relying entirely on comprehensive profiling. Using our leakage model, the executed attack demonstrated key recovery with fewer traces and less preparation than what is required for conventional profiled attacks. Additionally, we identified the specific conditions that allow the unique leakage characteristics of the screaming channels in tinyAES to emerge, based on our analysis of the assembly code.

Keywords: Side-channel attacks, Electromagnetic side-channels, Non-profiled attacks, Mixed-signal chip

1 Introduction

Mobile internet technology has been rapidly advancing, and devices equipped with wireless capabilities have become essential to our daily lives. As various devices communicate wirelessly in diverse locations, it is critical to incorporate robustness against the potential risk of various attacks. In particular, since adversaries can easily approach mobile devices physically, improving security against physical attacks is essential. Since the reported work by Kocher et al. [1], side-channel attacks have become of interest to a lot of researchers as a threat to cryptographic implementations [2, 3, 4, 5, 6]. The victim devices of these attacks include smartphones and IoT devices [7, 8, 6]. Different methodologies [9, 10, 11, 12], and implementations [13, 14, 15, 16] have confirmed the existence of electromagnetic side-channel attacks. In the context of low-power devices, these electromagnetic leakages are weak due to their limited strength and restricted measurement distance of a few millimeters.

Camurati et al. [7] presented screaming channels, through which an attacker can retrieve sensitive information at a significant distance from particular mixed-signal chips. The screaming channel attacks achieved key recovery against AES-128 in tinyAES [17], an unprotected software cryptographic implementation, from a distance of 10 meters. This approach has an advantage over traditional power and electromagnetic side-channel methods as it reduces the need for the attacker to collect traces in close proximity to the target device. Subsequent research by Camurati et al. [8] demonstrated that the leakage properties of the screaming channels diverge from those of electromagnetic side-channels. Profiled correlation attacks could successfully retrieve keys from distances up to 15 meters under more practical circumstances. Wang et al. [18, 19] introduced deep learning-based attacks that utilized screaming channel traces to enhance the efficiency of attacks. These attacks also retrieved keys without averaging traces with the same pairs of plaintext and key inputs.

Screaming channel attacks typically require an extensive number of traces for device profiling due to the unknown nature of screaming channel leakage models. Our interest is the potential existence of a simple leakage model for the screaming channel attacks. This work defines a novel leakage model, the MSB4 model, derived from our observations. The MSB4 model is unique because the leakage depends on the upper 4 bits of the intermediate value. To demonstrate the effectiveness of the MSB4 model, we carry out 2nd and 10th round attacks on tinyAES, using the MSB4 model to retrieve the keys. These attacks do not require comprehensive profiling, as previous studies suggested [8, 18, 19]. This unique aspect of the MSB4 model significantly reduces the complexity of screaming channel attacks. Our 2nd round attack is successful on the BLE Nano V2 and nRF DK boards with different supplementary components. This work broadly explores and discusses the conditions necessary for the MSB4 model by modifying the encryption code, and discovers that the MSB4 model could become applicable depending on the memory type or bus.

The main contributions of this study include:

- The discovery of distinguishable features in the upper 4 bits of the S-box input (the MSB4 model) through observations from our profiling, the proposal and verification of an attack using the MSB4 model, and the demonstration of the feasibility of key recovery on different devices.
- Discussion on the circumstances under which the MSB4 model emerges from the assembly code, with several conditions identified as influencing factors.

The paper is structured as follows: Section 2 provides an overview of previous screaming channel research. Section 3 defines the leakage model based on observed screaming channel leakage profiles. Section 4 details the attack methodology using the proposed model, along with the results of attack experiments. Section 5 discusses the conditions under which our leakage model arises. Finally, Section 6 concludes the paper.

2 Background

This section provides previous studies on screaming channels, as well as on leakage modeling for processor instructions.

2.1 Screaming Channels

Screaming channels, a type of electromagnetic side-channels introduced by Camurati et al. [7], enable the detection of leakage over long distances. They demonstrated key recovery against tinyAES and mbedTLS [20]. The leakage of screaming channels is visible in chips where noise-generating and noise-sensitive components are not appropriately separated, with previous research mainly focusing on mixed-signal chips. These chips, an example of which is the nRF52832 by Nordic Semiconductor, are a type of System-on-Chip (SoC). that integrates digital and analog circuits on a single silicon die.

Electromagnetic leakage is based on near-field emissions, which have low signal intensity and thus require close proximity for detection. This limits the feasibility of attacks to scenarios where the attacker has physical access to the target device. In contrast, screaming channel leakage is caused when the electromagnetic leakage in the digital circuit is propagated to the wireless communication component. These leakages are then incorporated into analog parts and transmitted as radio waves, enabling leakage detection from a much greater distance. This significantly expands the range of possible attack scenarios, posing a greater threat to cryptographic implementations.

Concerning the leakage model, prior research [8] targeted tinyAES on the nRF52832 chip. Figure 1 shows the results of this profiling utilizing the Hamming weight (HW) model of the S-box output as the leakage model, defined as:

$$model_{HW}\left(p,k\right) = HW\left(S\left[p \oplus k\right]\right), \tag{1}$$

where the function $S[\cdot]$ returns a value using the AES S-box table, while $HW(y)$ returns the Hamming weight for the intermediate value y. The blue line shows the profile of electromagnetic side-channels, while the orange line shows the profile of screaming channels. Although the electromagnetic leakage can be represented by the Hamming weight model, it has been determined that the screaming channel cannot be represented by this model. Moreover, the linear regression model for the S-box output, which considers bit weights, exhibited a lower correlation with the profile compared to the electromagnetic side-channels. This may be due to thermal noise distorting leakage from the target to the attacker, as well as intervals of inactivity during packet transmission.

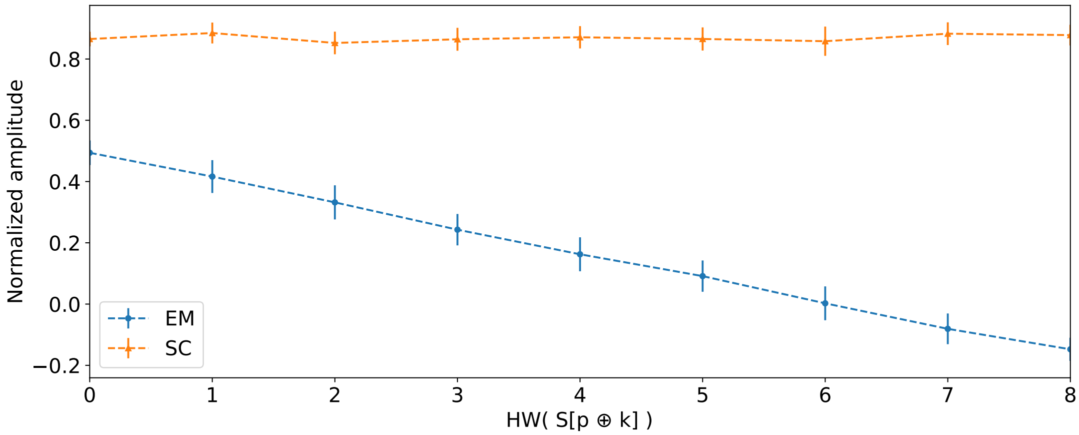


Figure 1: Difference between electromagnetic (EM) and screaming channel (SC) leakage. To compute the profiles, we used publicly available trace sets and programs [21].

Danieli et al. [22] observed changes in the spectrogram to pinpoint the on-chip regions, such as memory, universal asynchronous receiver transmitter (UART), and SPI Flash, where screaming channel leakage occurs. Their findings suggested that the leakage is not limited to the CPU but also originates from memory accesses and UART communication. While electromagnetic leakage from the SPI Flash, an off-chip component, was detected, screaming channel leakage was not detected at a distance of 10 cm.

Camurati et al. also conducted profiled correlation attacks on the S-box input utilizing screaming channels [8]. They accomplished full key recovery with 5000 profiling traces and 1273 attack traces at a distance of 10 cm. The furthest successful attack distance recorded was 15 m. However, the high clock frequency of the low-power target device posed significant sampling rate constraints, making the extraction of numerous potential leakage sample points challenging.

Previous research [8, 18, 19] predominantly employed profiled attacks. These attacks are highly efficient in key recovery, but they require the attacker to have access to a reference device for profiling. Conversely, non-profiled attacks execute correlation attacks based on general leakage characteristics knowledge. Given that the Hamming weight leakage model, which was applicable in electromagnetic side-channels, is unsuitable, profiling becomes essential. This research introduces a simplified model of the screaming channels leakage on tinyAES, demonstrating its potential utility for attacks.

2.2 Instruction-Level Leakage Profiling

A method to model leakage involves considering leakage that is dependent on processor instructions. This strategy requires less knowledge of the attacker's hardware than circuit-level modeling using netlists. However, it requires more knowledge of the processor than black-box modeling approaches such as using intermediate value leakage features in cryptographic implementations.

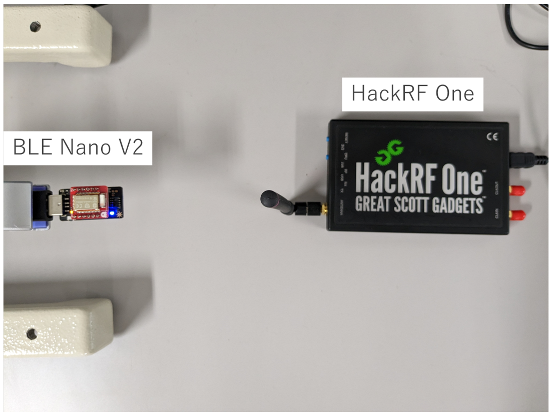


Figure 2: Experimental equipment.

McCann et al. [23] created a power or electromagnetic model for the operand values and transitions of each assembly instruction in the Arm Cortex-M series. They conducted an F-test on the models to investigate the contribution of operand values or transitions to leakage. They observed, for instance, that the first operand value of ALU instructions (such as add, sub, etc.) and the second operand value of the load instruction, both contributed to leakage in the Arm Cortex M4.

Barenghi et al. [24] derived detailed microarchitectural characteristics, such as pipeline processing, from the cycle count per instruction and deduced potential leaking units based on these characteristics. They discovered leaks in the IS/EX pipeline registers, load units, and register files. Specifically, they found that pipeline registers and load units were involved in the leakage, even when consecutive instruction sequences accessed different registers.

3 Exploring Leakage Characteristics of Screaming Channels

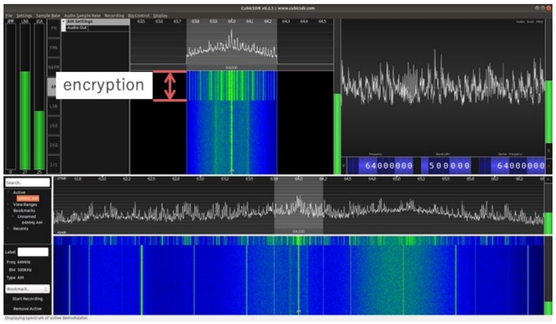
This section introduces the observations made from several profiles. Based on these observations, this study proposes alternative leakage models of the screaming channels. These profiles are created from two sources: traces publicly available by Camurati et al., and traces obtained from our experimental setup.

3.1 Experimental Environment

In this work, the experimental setup was constructed based on the trace collection and key recovery environment developed by Camurati et al. [8]. Figure 2 shows the experimental equipment, with the BLE Nano V2 located on the left and the Hack RF One [25] on the right. Both are operated from the same laptop. The distance between the target chip and the collection device for the traces publicly available by Camurati et al. is 10 cm. In contrast, the traces from our experimental setup were obtained at a distance close enough to be in physical contact (0 cm). This section describes the target device and the collection device, along with their setups, as well as the preprocessing of traces for analysis.

3.1.1 Target device

The target device is Red Bear's BLE Nano V2 [26], which features Nordic Semiconductor's nRF52832 [27] mixed-signal chip. The device operates at a clock frequency of 64 MHz. This work focuses on attacking the cryptographic implementation, tinyAES. The Particle debugger was utilized for writing



Freq 64MHz
BW 500KHz
Type AM

64000000 500000 64000000

Displaying spectrum of active demodulator.

Figure 3: Signal during encryption (CubicSDR). The spectrogram shows different patterns only during encryption.

to the target device and transmitting the cryptographic commands. The device supports Bluetooth Low Energy (BLE) communication, offering a bandwidth from 2.4000 to 2.4835 GHz. Typically, it hops between 40 different channels via frequency hopping. However, to simplify our investigation, we set the radio carrier frequency to a constant 2.400 GHz.

3.1.2 Measurement setup

The traces were collected using a Hack RF One, which is a software-defined radio (SDR), at a frequency of 2.528 GHz, which is the sum of the radio carrier frequency and an integer multiple of the target chip clock frequency. By examining the spectrogram during encryption, it was possible to capture the leakage, as evidenced by the presence of bands in the spectrogram, as shown in Figure 3. To achieve better measurements, it is necessary to tune the gain of the software-defined radio.

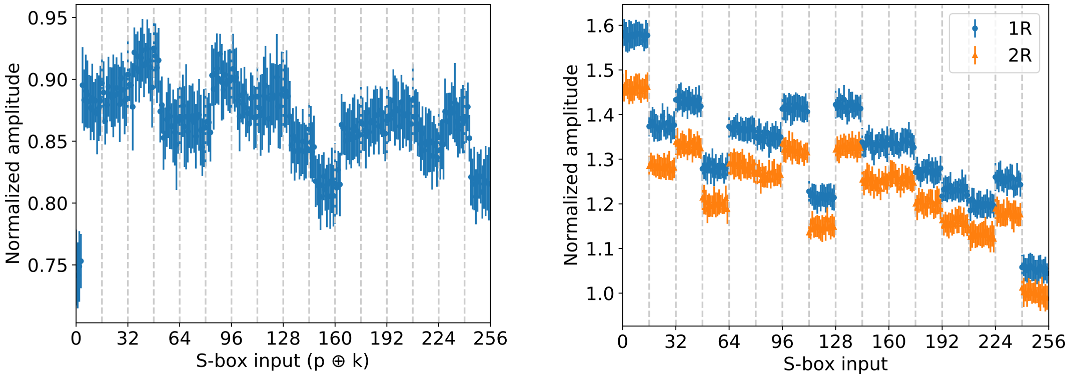
3.2 Identification of Leakage Characteristics through Profiles

Figure 4a illustrates the profile of the screaming channel leakage in tinyAES, built from traces made publicly available by Camurati et al. [8]. The figure shows the relationship between the S-box input values $p \oplus k$ and the radio signal is shown as mean and variance. Notably, the amplitude is lower when the input values for the S-box are 0, 1, 2, and 3. Furthermore, the amplitude does not solely depend on the number of ones in the 8-bit S-box input, but it's observable that the amplitude is separated into groups. For instance, the plots of mean and variance for the intermediate value ranges $[128, 143]$ and $[144, 159]$ appear visually separated, respectively. In combination with Figure 1, this implies that the Hamming weight model is not suitable as a leakage model for both the input and output of the S-box of tinyAES.

Figure 4b illustrates the profiles of the S-box input value for the 1st and 2nd rounds with respect to subkey k_0; these profiles were generated using traces collected at our experimental setup. k_i denotes the i-th subkey. An observation of these profiles reveals an apparent separation into recognizable groups. In the traces collected at our experimental setup, inserting a reference line every 16 units on the profiles results in such segmentation. This pattern suggests a dependency of the leakage on the upper 4 bits of the S-box input value. Consequently, we called the leakage model for the screaming channel the MSB4 model.

By focusing only on the leakage model derived from our profile, an alternative interpretation of the MSB4 model can be deduced. The leakage might be affected by the count of ones in the upper 4 bits of the intermediate value, indicating that it can be represented by Hamming weights. Consequently, this study defines the Hamming weights of the upper 4 bits of the intermediate value as an additional leakage model and then validates its applicability in the attack.

Figures 4a and 4b lead to the following observations.



(a)

(b)

Figure 4: Profile built for S-box input in the 1st round using the traces publicly available by Camurati et al. [21] (a). Comparison of profiles built for S-box input in the 1st and 2nd rounds with the traces collected at our experimental setup (b). Dashed lines, drawn at intervals of 16 along the horizontal axis, effectively illustrate the segmentation of the block.

Grouping based on the upper 4 bits of S-box input In both experimental environments, the leakages are separated into groups based on the upper 4 bits of the S-box input. Specifically, Figure 4b reveals the presence of 16 groups, each consisting of 16 bits, representing intermediate values.

Dependency on setup The magnitude and order of the groups differ between the experimental setups. In Figure 4b, the group associated with the smallest S-box input demonstrates the strongest leakage, while this relationship differs in Figure 4a.

Similarity across rounds Our experimental setup demonstrates significant similarity in the leakage models between the 1st and 2nd rounds. In Figure 5, although specific signal strengths vary between rounds, the overall relationship remains highly similar, with only a vertical shift.

Utilizing the traces publicly available by Camurati et al., this study observed a characteristic that the screaming channel leakages are separated into groups based on the upper 4 bits of S-box input values. However, this characteristic was not replicated when profiling the electromagnetic leakage at the same intermediate values. Building on our observations and previous research [8], a distinction from the electromagnetic leakage model becomes evident. This finding suggests that the MSB4 model is unique to screaming channels.

4 New Attacks Based on Discovered Leakage Characteristics

Although the MSB4 model focuses on only a part of the intermediate value, it presents unique attributes distinct from conventional Hamming weight models. Leveraging these attributes, we propose two attacks against tinyAES and validate their effectiveness. The first method is a 2nd round attack that focuses on the leakage depending on the upper 4 bits of the intermediate values. The second method is a 10th round attack that formulates a leakage model that combines the MSB4 model and Hamming weight, employing the correlation attack in the process. Although not explored in this study, it's worth noting that there are other non-profiled attacks [11, 12], including collision attacks.

4.1 Second Round Attack

The MSB4 model, given its inability to establish the group's magnitude and order, fails to yield key recovery when a non-profiled 1st round attack is attempted. This model requires a nonlinear operation to impact all bits of the intermediate value from the upper four key-guess bits. As such, key recovery may only be feasible from the 2nd round onward. In addition, the complexity of key candidates poses a challenge when launching a 2nd round attack. Therefore, for an attack leveraging the MSB4 model, a suitable distinguisher is required to reduce the complexity of key candidates.

4.1.1 Chosen plaintext attack with reduced complexity

When considering a 2nd round attack, a random plaintext attack requires 2^{32} key candidates to execute the MixColumns operation in the 1st round. Conversely, a chosen plaintext attack reduces the key candidates to 2^8 by only changing the byte corresponding to each subkey and keeping the other bytes constant. We assume that the attacker has the capability to repeatedly encrypt the same plaintext, irrespective of its randomness. This implies that the attacker has the ability to encrypt any chosen plaintext. Given the consistent assumptions about the attacker, this study performed a chosen-plaintext attack on the 2nd round S-box input.

4.1.2 Variance-based distinguisher

Each group's range of amplitudes is much smaller compared to the range of amplitudes obtained for all intermediate values. Therefore, the distinguisher employed is variance, which is smaller for correct keys and larger for incorrect keys and points unrelated to sensitive information.

4.1.3 Attack procedure

A set of plaintext, denoted as $\mathcal{P}_i$, comprises four ordered tuples associated with the subkey k_i. Among these tuples, one element ranges from 0 to 255, while the remaining elements are held constant at 0, such that

$$\mathcal{P}_i = \{(p, 0, 0, 0)\}. \tag{2}$$

The index of the variable element within each tuple corresponds to the row number of the targeted byte in the attack. Take subkey k_{10} as an example. Given that it is on the third row, the variable is the third element, therefore $\mathcal{P}_{10} = \{(0, 0, p, 0)\}$. A key-guess tuple, $\mathcal{K}_i$, is defined similarly for subkey k_i. The leakage model at the S-box input of the 2nd round using the MSB4 model for key guess k_0 is then defined.

The MSB4 model for key guess k_0 is defined as follows:

$$model_{2R,MSB4}(p, k) = MSB4 \left[MC \begin{bmatrix} S[p \oplus k] \\ S[0 \oplus 0] \\ S[0 \oplus 0] \\ S[0 \oplus 0] \end{bmatrix} \right]. \tag{3}$$

Here, $MSB4[\cdot]$ works as a filter that passes only the upper 4 bits of the 8-bit elements. The calculation of the 1st round AddRoundKey can be omitted, as the model does not differentiate between the orders of each class. As AddRoundKey is a linear operation, and the round key remains fixed for a given key, adding these fixed values alters the classes, but only shifts the value of each group by a constant amount.

The trace set, denoted as L, was subdivided into a subset L_y, with this classification conducted in accordance with the leakage model, $y = model_{2R,MSB4}$. After categorizing the traces, compute the variance within each class. The aggregate variance $VAR[\cdot]$ across the tuples for key guess k_i is assigned as the score $\hat{s}_{cp}(k_i)$:

$$\hat{s}_{cp}(k_i) = \Sigma_0^3 \Sigma_0^c VAR[L_y], \tag{4}$$

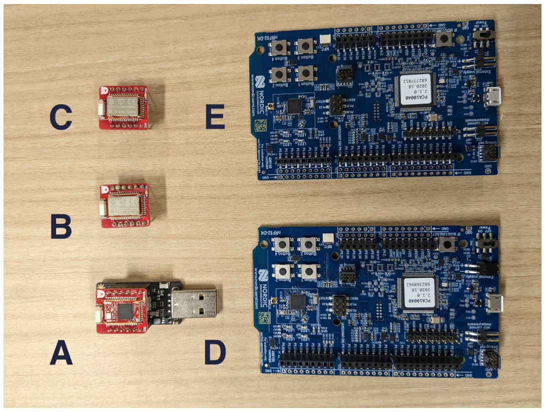


Figure 5: Target devices. Devices A, B, and C are BLE Nano V2. and Devices D and E are PCA10040.

Table 1: Target Device Specifications with nRF52832 Chip.

Device	Name	Shield	Antenna
BLE Nano V2	A	Unshielded	Chip antenna
BLE Nano V2	B, C	Shielded	Chip antenna
PCA10040	D, E	Unshielded	PCB trace antenna

where c denotes the number of classes in the leakage model. Finally, the key guess k_i that returns the minimum score, $\hat{s}_{cp(k_i)}$, is selected as the correct key.

Our described 2nd round attacks present a novel direction in screaming channel attacks, maintaining the attacker assumptions of preceding research while significantly reducing the necessary initial setup.

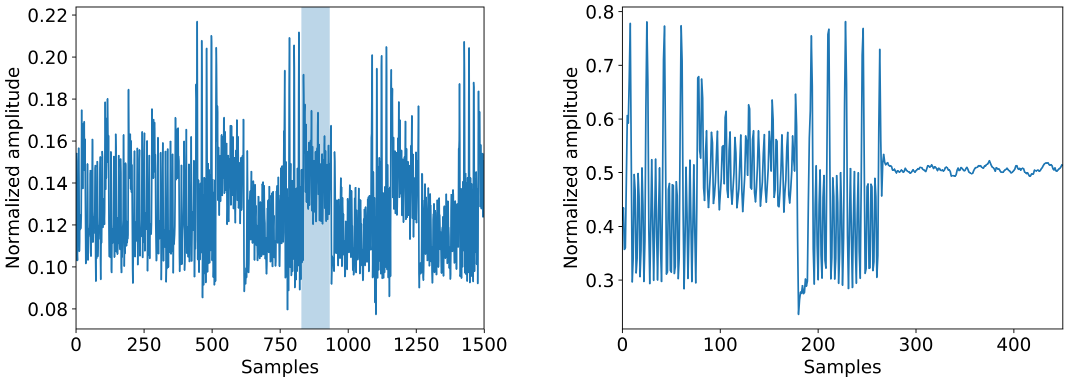
4.1.4 Experimental setup

The experimental setup was established and traces were collected in accordance with the procedure outlined in Section 3.1. The distance between the target and the software-defined radio is 10 cm. This experiment used five targets as shown in Figure 5 and Table 1. The PCA10040 [28], a development kit board for the nRF52 series, has a different transmission antenna compared to the BLE Nano V2. Notably, any trace with significant high noise levels was excluded from the analysis, as a single excessively noisy trace can impact the overall attack performance due to the reliance on variance.

Figure 6a presents a representative trace collected, highlighting the range of sample points utilized in the trace, specifically during the SubBytes and ShiftRows operations. This trace, recurring in line with the number of AES rounds, signifies the AddRoundKey operation because a segment from 400 to 500 repeats 11 times. It also represents the MixColumn operation as the trace from 620 to 740 in the 1st round was not observed in the AddRoundKey of the 9th round and the 10th round trace depicted in Figure 6b. Consequently, the remaining range is deduced to include the SubBytes and ShiftRows operations.

4.1.5 Second round attack result

Table 2 demonstrates the successful key recovery from all devices in the 2nd round attack. For comparison, it shows the key recovery results of the previous study and the correlation radio analysis (CRA) to 1st round with 50000 traces collected in our environment. Furthermore, it indicates that a chosen



(a)

(b)

Figure 6: Range of sample points to 2nd round attack (a). Trace used for the 10th round attack (b).

Table 2: The number of traces required for full key recovery from each of the five devices (devices A-E). Distance to target 10 cm.

Attack method	Device (name)	Attack traces (profiling)
Previous work [8]	BLE Nano V2	1273 (5000)
CRA	BLE Nano V2 (A)	> 50000
2nd round attack	BLE Nano V2 (A)	2896
2nd round attack	BLE Nano V2 (B)	1680
2nd round attack	BLE Nano V2 (C)	1520
2nd round attack	PCA10040 (D)	3616
2nd round attack	PCA10040 (E)	1776
10th round attack	BLE Nano V2 (A)	175
10th round attack	BLE Nano V2 (B)	34
10th round attack	BLE Nano V2 (C)	69
10th round attack	PCA10040 (D)	> 5000
10th round attack	PCA10040 (E)	900

plaintext attack in the 2nd round using the MSB4 model can be executed regardless of distinct transmission antennas. The varying number of traces needed for the attack across devices can be attributed to the fact that noise greatly impacts the key distinguisher by variance. Figure 7 illustrates the scores for each subkey candidate, with the lowest scores associated with the correct key, particularly notable for k_4 and k_{13}, which are apparently distinct from incorrect keys. These findings suggest that this attack method can be employed on other devices without profiling.

4.2 Tenth Round Attack

Based on the approach outlined in Section 3.2, the profiles generated in our experiment not only classify the upper 4 bits of the S-box input values but can also approximate the Hamming weight of these upper 4 bits. This feature was used to execute the 10th round attack.

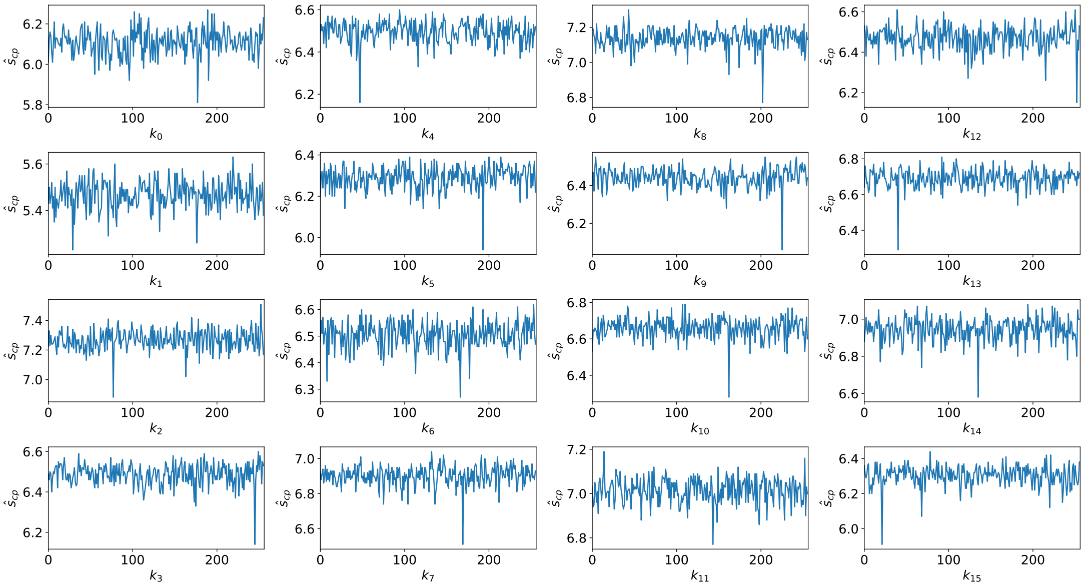


Figure 7: Scores for each key candidate by the 2nd round attacks using 250 traces.

4.2.1 Tenth round attack method

The attack method we implemented was correlation radio analysis, utilizing random ciphertexts. The correlation attack technique operates by finding the key guess with the highest correlation coefficient. This is obtained by correlating the leakage model and the leaked information gathered from the side-channel for each key candidate.

This work represents C as the set of ciphertexts, RK as the set of 10th round keys, and L as the set of leakages collected during encryption. For each key candidate $rk \in RK$, the predicted leakage X_{rk} is generated as

$$X_k = \{x = model_{MSB4,10R}\left(c, rk\right), c \in C\}\,. \tag{5}$$

The leakage model is defined as:

$$model_{MSB4,10R}(c, rk) = HW(MSB4[S^{-1}[c \oplus rk]]). \tag{6}$$

The score $\hat{s}_{rk}$, used for key recovery, is the Pearson's correlation coefficient between the leakage L and the predicted leakage X_{rk}, represented as

$$\hat{s}_{rk} = \rho\left(L, X_{rk}\right) \;= \frac{\sum_{l \in L, x_{rk} \in X_{rk}} \left(l - \bar{l}\right)\left(x_{rk} - \bar{x}_{rk}\right)}{\sqrt{\sum_{l \in L}\left(l - \bar{l}\right)^2 \sum_{x_{rk} \in X_{rk}}\left(x_{rk} - \bar{x}_{rk}\right)^2}}, \tag{7}$$

where $\bar{l}$ and $\bar{x}_{rk}$ are the means of l and x_{rk}, respectively. The score $\hat{s}_{rk}$ is determined for all key candidates rk in the key set RK. Eventually, the key guess with the highest score is determined to be the correct subkey and the original key is acquired by using the inverse function of the key schedule function.

4.2.2 Tenth round attack result

As demonstrated in Table 2, the 10th round attack successfully retrieved keys from all devices except device D. For devices $A - C$, through the application of an appropriate leakage model, this attack achieved key recovery using fewer traces than in the previous fully profiled attack. The number of traces needed for the attack varied significantly between the BLE Nano V2 and PCA10040, with device D not being able to fully recover the key even with 5000 traces, resulting in an average key rank of 1.8 for the subkey.

The nRF52 DK board required a greater number of traces for the 10th round attack compared to the 2nd round attack. This was due to the lack of an appropriate leakage model that exclusively depends on the upper 4 bits of the S-box input value and does not correlate with its Hamming weight. Consequently, the leakage only depends on the upper 4 bits of the S-box input value and does not depend on its Hamming weight.

5 Discussion

This section discusses the conditions under which the MSB4 model can be applied. Specifically, we suggest that the applicability of the MSB4 model is notably influenced by the memory type and address.

5.1 Related Work on Leakage Regions

Our claims are consistent with existing work; for instance, Camurati et al. [7, 8] attacked two different implementations of AES. They were able to recover keys for mbedTLS using non-profiled attacks with the Hamming weight model, whereas key recovery in tinyAES always required a profile. These findings suggest that the complexity of the leakage model may vary depending on the specific software implementation. On the other hand, Mac et al. [23] found that in the Arm Cortex-M4, a processor also used in the nRF52832 chip, leakage is associated with the value of the second operand indicating the memory address. Barenghi et al. [24] previously noted that transitions between memory addresses and load destination registers leaked during IS/EX pipeline register transitions and consecutive load instructions.

5.2 Leakage Characteristics for Implementations Difference

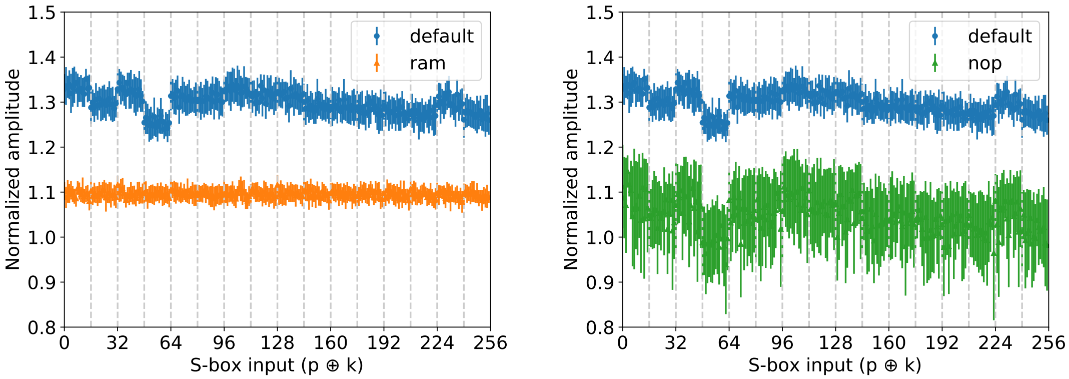
The principal distinction between the tinyAES and mbedTLS implementations lies in whether they are simple AES implementations or utilize T-tables. In T-table implementations, SubBytes, ShiftRows, and MixColumns are jointly processed every 4 bytes using a T-table. mbedTLS offers two methods: storing the T-table in ROM in advance or generating it just before encryption. However, our primary focus is the difference in memory placement for the nonlinear table between tinyAES and mbedTLS. In tinyAES, the S-box is stored in the Flash area, while in mbedTLS, the T-table resides in the data RAM area.

We hypothesized that the leakage model varied depending on the location of the nonlinear table. Therefore, we relocated the S-box in tinyAES from the Flash area to the data RAM area, intending to observe any potential changes in leakage models. Compared to the spectrogram observations by Danieli et al. [22], this approach allows us to more concretely demonstrate conditions under which attackers can obtain sensitive information through profiling. Additionally, it enables discussions regarding leakage from the on-chip Flash area in mixed-signal chips. The distance between the target and the software-defined radio was set to 10 cm, and traces were collected for each implementation. Our profiling of device A's leakage, as shown in Figure 8a, revealed no apparent features of the MSB4 model. Additionally, we conducted a non-profiled correlation radio attack on tinyAES, testing various S-box arrangements. With the standard tinyAES, we were unable to recover even a single byte in 50000 traces. However, the tinyAES version with a modified S-box arrangement allowed the recovery of all subkeys within the same number of traces.

This work also examined the effects of unnecessary operations identified in the assembly of the SubBytes process. These include excessive duplication of register values that hold the intermediate values, such as $p \oplus k$ and $S[p \oplus k]$, and unnecessary memory reads and writes of the intermediate values. This work eliminated these procedures, replacing them with 'nop' instructions. If these unwanted

Table 3: Assembly code to get S-box output.

Address	Binary code	Assembly code	Comment
0x7830	0x4a03	ldr r2, [pc, #12]	[pc, #12] = &S[0]
0x7832	0x5cd3	ldrb r3, [r2, r3]	r3 = p ⊕ k



(a)

(b)

Figure 8: (a): Profile of screaming channel trace during encryption in the case of standard tinyAES and when the S-box table is placed in RAM. (b): Profile when unnecessary assembly code is converted to 'nop' instructions.

processes encompass the conditions under which the MSB4 model emerges, the features of the MSB4 model would be expected to disappear in the profile of the modified AES leakage. The profiling results shown in Figure 8b indicate several groupings that appear to be consistent with the MSB4 model.

Based on these results, we confirmed that the appearance conditions of the MSB4 model are affected by the type and address of the memory. This provides an important guideline for successful attacks using the MSB4 model.

To better clarify our findings, it's necessary to understand the architecture of some commonly used devices. The nRF52832 features an on-chip flash memory, which is exclusively readable by the CPU. In contrast, the data RAM area can be both read and written by other peripherals, utilizing the easy-to-use direct memory access module (EasyDMA). Such architectural differences can greatly influence leakage characteristics. We believe that the leakage following the MSB4 model obtained in our experiments originated from the on-chip flash. Leakage detection from the off-chip flash region via screaming channels was limited and could not be detected at 10 cm [22]. Our findings suggest that leakage from on-chip Flash, where access from other peripherals is restricted, might have a more dominant effect on the leakage characteristics. Nonetheless, empirical studies on alternative chips are essential to corroborate this hypothesis, and why specific leakage models appear needs to be clarified in future studies.

6 Conclusions

Our study investigated the leakage characteristics of screaming channels. In particular, it highlighted more appropriate leakage characteristics compared to the conventional Hamming weight model. Profiling on the BLE Nano V2 revealed noticeable leakage characteristics in the upper 4 bits of the S-box input in tinyAES. This finding allowed us to provide a screaming channel attack on tinyAES without the need for extensive profiling. The efficacy of our leakage model is apparent as it decreases the typical traces and preparatory work required in profiled attacks while enabling on-the-fly key recovery. Moreover, our work explored the circumstances under which our model emerges from the assembly code and suggested that the memory placement of nonlinear tables could be one of these conditions.

Future work involves statistically demonstrating the assembly instructions that impact leakage, as well as investigating the bounds of cryptographic implementations, particularly those with countermeasures, where the MSB4 model could be utilized.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number 20K19798.

References

[1] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual international cryptology conference*, pages 388–397. Springer, 1999.

[2] C. Archambeau, E. Peeters, F. X. Standaert, and J. J. Quisquater. Template attacks in principal subspaces. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4249 LNCS:1–14, 2006.

[3] François Durvaux and François Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9665:240–262, 2016.

[4] Owen Lo, William J. Buchanan, and Douglas Carson. Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). *Journal of Cyber Security Technology*, 1(2):88–107, 2017.

[5] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8616 LNCS(PART 1):444–461, 2014.

[6] Alexander S La Cour, Khurram K Afridi, and G Edward Suh. Wireless charging power side-channel attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 651–665, 2021.

[7] Giovanni Camurati, Sebastian Poeplau, Marius Muench, Tom Hayes, and Aurélien Francillon. Screaming channels: When electromagnetic side channels meet radio transceivers. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 163–177, 2018.

[8] Giovanni Camurati, Aurélien Francillon, and François-Xavier Standaert. Understanding screaming channels: From a detailed analysis to improved attacks. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 358–401, 2020.

[9] Dakshi Agrawal, Josyula R Rao, and Pankaj Rohatgi. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems-CHES 2003: 5th International Workshop, Cologne, Germany, September 8–10, 2003. Proceedings 5*, pages 2–16. Springer, 2003.

[10] David P. Montminy, Rusty O. Baldwin, Michael A. Temple, and Mark E. Oxley. Differential electromagnetic attacks on a 32-Bit microprocessor using software defined radios. *IEEE Transactions on Information Forensics and Security*, 8(12):2101–2114, 2013.

[11] Amir Moradi, Oliver Mischke, and Thomas Eisenbarth. Correlation-enhanced power analysis collision attack. In *Cryptographic Hardware and Embedded Systems, CHES 2010: 12th International Workshop, Santa Barbara, USA, August 17-20, 2010. Proceedings 12*, pages 125–139. Springer, 2010.

[12] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In *Proceedings of the 2016 ACM Workshop on Theory of Implementation Security*, pages 5–15, 2016.

[13] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM Side-Channel(s). *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2523:29–45, 2003.

[14] Vincent Carlier, Hervé Chabanne, Emmanuelle Dottax, and Hervé Pelletier. Electromagnetic side channels of an FPGA implementation of AES. *Cryptology Eprint Archive*, 2004.

[15] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2162:251–261, 2001.

[16] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. ECDH key-extraction via low-bandwidth electromagnetic attacks on PCs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9610:219–235, 2016.

[17] kokke. tiny-AES-c: Small portable AES128/192/256 in C. https://github.com/kokke/tiny-AES-c, 2019. Accessed: 2023-07-07.

[18] Ruize Wang, Huanyu Wang, and Elena Dubrova. Far field EM side-channel attack on AES using deep learning. In *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*, pages 35–44, 2020.

[19] Ruize Wang, Huanyu Wang, Elena Dubrova, and Martin Brisfors. Advanced far field EM side-channel attack on AES. In *Proceedings of the 7th ACM on Cyber-Physical System Security Workshop*, pages 29–39, 2021.

[20] Mbed TLS. https://tls.mbed.org/, 2017. Accessed: 2023-07-07.

[21] EURECOM S3 Group. Screaming channels open-source code. https://eurecom-s3.github.io/screaming_channels/, 2018.

[22] Erez Danieli, Menachem Goldzweig, Moshe Avital, and Itamar Levi. Revealing the Secrets of Radio-Enabled Embedded Systems: on extraction of raw information from any on-board signal through RF. *Cryptology ePrint Archive*, 2023.

[23] David McCann, Elisabeth Oswald, and Carolyn Whitnall. Towards practical tools for side channel aware software engineering: 'Grey box' modelling for instruction leakages. *Proceedings of the 26th USENIX Security Symposium*, pages 199–216, 2017.

[24] Alessandro Barenghi, Luca Breveglieri, Niccolo Izzo, and Gerardo Pelosi. Exploring Cortex-M Microarchitectural Side Channel Information Leakage. *IEEE Access*, 9:156507–156527, 2021.

[25] nRF5x/nRF52832/docs/Specifications.md at master · redbear/nRF5x · GitHub. https://github.com/redbear/nRF5x/blob/master/nRF52832/docs/Specifications.md.

[26] nRF5x/nRF52832/docs/Specifications.md at master · redbear/nRF5x · GitHub. https://github.com/redbear/nRF5x/blob/master/nRF52832/docs/Specifications.md.

[27] nRF52832 Product Specification. https://infocenter.nordicsemi.com/pdf/nRF52832_PS_v1.8.pdf.

[28] nRF5x/nRF52832/docs/Specifications.md at master · redbear/nRF5x · GitHub. https://github.com/redbear/nRF5x/blob/master/nRF52832/docs/Specifications.md.