The Weighted Sum Correlation Analysis

Elena Dubrova, Sönke Jendral, Yanning Ji and Ruize Wang
KTH Royal Institute of Technology, Stockholm, Sweden
{dubrova,jendral,yanning,ruize}@kth.se

Abstract

This paper introduces the weighted sum correlation analysis method, a profiled higher-order side-channel attack that quantifies the significance of time-domain samples based on a chosen leakage model. We also demonstrate the utility of the Hilbert transform in side-channel analysis, showing how its phase-shifting property can be exploited to construct an effective fused score that combines multiple correlation coefficients into a single metric. We validate the presented method on the challenging case of the AES-CCM accelerator in a commercial Bluetooth chip, leveraging RF signals captured via a software-defined radio as a side channel. Compared to the correlation analysis methods presented at RWC'25 and CHES'25, the weighted sum approach achieves at least a threefold reduction in the number of traces required for key recovery. Remarkably, it also outperforms deep learning-based analysis.

1 Introduction

Redundancy is a well-known method for tolerating faults. In the field of fault-tolerant computing, the information is duplicated in space (extra components) or time (re-computation) to detect faults, or triplicated to correct errors [Dub13]. Additionally, the information may be diversified to mitigate common-mode faults which affect multiple redundant units in the same way. For instance, instead of comparing two identical modules, a modules and its complemented version may be used. Common-mode faults are caused by phenomena that simultaneously affect all redundant units due to shared resources such as a common communication bus or power supply, or shared environmental conditions. In the context of side-channel analysis (SCA), avoiding common-mode faults is clearly desirable because repeated measurements affected by the same fault do not bring much extra information.

In this paper, we incorporate several types of redundancy into the SCA in order to make the analysis more robust. We build upon the existing knowledge in the field and add some new insights. The known redundancy techniques which we employ are:

1. Using both amplitude and phase components of a complex-valued signal, proven to be useful in several studies utilizing a software-defined radio (SDR) or a comparable radio frequency (RF) front-end for data acquisition [O4, AHC+25].

2. Utilizing multiple leakage models, such as both the Hamming weight and the Hamming distance, which helps decrease the number of traces required for the attack [WD21].

3. Applying a combining function to fuse the information extracted from multiple sources, a technique employed at various levels (data, feature, decision) in [ARR03, MOW14, YZC+17, GMGH19, HWW22].

Two main novel contributions of this paper are:

1. Extending the classical correlation power analysis (CPA) [BCO04] into a profiled attack called the weighted sum correlation analysis[1]. In this attack, at the profiling stage, the significance of each data sample of a time-domain trace is quantified with respect to a chosen leakage model based on traces collected from profiling devices. Unlike deep learning-based or template attack profiling methods,

[1]Since the underlying statistical approach can be applied regardless of the physical channel through which the leakage is observed, we prefer to use the more general term *correlation analysis* rather than CPA.

which learn/create a template for each value of the subkey, the presented profiling method aims to identify samples with higher average leakage, irrespective of the subkey value.

2. Demonstrating the usefulness of the *Hilbert transform* in SCA. Together, the original real-valued signal and its transformed counterpart offer complementary insights into the underlying information leakage. The Hilbert transform rotates the phase of each frequency component of the original signal by 90° without affecting its magnitude. We show how this property can be utilized to construct an effective fused score that combines multiple correlation coefficients into a single metric.

We validate the weighted sum method on the challenging case of the AES-CCM accelerator in the commercial Nordic Semiconductor nRF52832 Bluetooth chip. This widely deployed chip is embedded in millions of devices globally, including wearables, smart home products, and industrial IoT applications. The first successful key-recovery attack on this AES accelerator was recently presented at RWC'25 [JDW25a, JDW25b]. Our work significantly benefits from the reverse engineering results disclosed in that study.

As in [CPM+18], we leverage side-channel information from AES computations that is unintentionally transmitted by the nRF52832 along with the RF signals. The RF signals are captured using an antenna positioned one meter from the device and demodulated at a specific center receiving frequency using an SDR.

We benchmark the weighted sum method against the RWC'25 CPA attack which uses only the amplitude component of a complex-valued signal and the Hamming weight leakage model, as well as against the CHES'25 CPA attack which uses both the amplitude and phase and the Hamming weight leakage model. Our experimental results demonstrate that the presented approach reduces the number of required attack traces at least by a factor of three compared to the CHES'25 and RWC'25 methods. This is not too surprising since CHES'25 and RWC'25 attacks are first-order attack while the presented attack is higher-order. More notably, the weighted sum approach outperforms deep learning-based SCA, which is also a higher-order attack.

The rest of the paper is organized as follows. Section 2 provides background information and introduces the notation used throughout the paper. Section 3 reviews previous work on related side-channel attacks. Section 4 defines the threat model. Section 5 describes the experimental setup. Section 6 explains the rationale for employing the Hilbert transform in SCA. Section 7 formulates the research problem, and Section 8 introduces the weighted sum CPA method as a proposed solution. Section 9 summarizes the experimental results. Section 10 discusses potential future work. Section 11 concludes the paper.

2 Background

This section presents background on side-channel attacks and sources of leakage in RF signals.

2.1 Side-Channel Attacks

Side-channel attacks, introduced by Kocher et al.[Koc96], have become a significant threat to the security of cryptographic algorithm implementations. Unlike conventional cryptanalysis, which targets the mathematical properties of the algorithm, side-channel attacks exploit information inadvertently leaked by physical devices during computation. Such leakage can be observed through various physical phenomena, including execution time [Koc96], power consumption [KJJ99], and electromagnetic (EM) emissions [AARR03]. By collecting side-channel data and correlating it with a hypothetical leakage model, an attacker may be able to recover secrets such as cryptographic keys, despite the theoretical strength of the underlying cryptographic algorithm.

Side-channel attacks can be classified into two categories: profiled and non-profiled.

2.1.1 Profiled and Non-Profiled Attacks

In *profiled* side-channel attacks, the attacker is assumed to have access to a device identical to the target device. The goal is to construct a leakage profile of the target device using side-channel information collected from the profiling device(s) [CRR03].

Profiled attacks consist of two stages: the *profiling stage*, during which traces are captured and a leakage profile is constructed; and the *attack stage*, in which the constructed leakage profile is applied to traces

collected from the target device to recover the secret. With the rapid advancement of artificial intelligence, deep learning (DL) has become an increasingly popular method for constructing leakage profiles for side-channel attacks [MPP16].

In contrast, *non-profiled* attacks are carried out without prior knowledge of the target device's leakage behavior, relying solely on traces collected during the attack phase. A widely used technique in this category is CPA which uses the Pearson correlation coefficient as a statistical distinguisher [BCO04].

2.1.2 Correlation Power Analysis

Correlation power analysis [BCO04] measures the linear correlation between observed data and hypothetical leakage values estimated using the selected leakage model. The Pearson correlation coefficient, r, between two datasets, $\{x_1, ..., x_n\}$ and $\{y_1, ..., y_n\}$, is computed as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{1}$$

where n is the sample size, x_i, y_i are the individual samples, and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ are the datasets means. CPA quantifies the extent to which the actual measurements correlate with the estimated hypothetical leakage values.

2.1.3 Deep Learning-Based Analysis

Deep learning-based SCA leverages DL algorithms to assess information leakage [MPP16, PSB+18]. In profiled DL-SCA [PH18], a DL model is trained on a labeled set of traces captured from a profiling device identical to the target device. The labels represent the values of certain key-dependent intermediate variables within the cryptographic algorithm implemented on the device, e.g., the first round S-Box output in AES.

A neural network $\mathcal{N}$ learns to map a trace $T \in \mathbb{R}^m$ into a score vector $S = \mathcal{N}(T) \in \mathbb{I}^n$, where m is the number of data points in T, n is the number of different labels (classes), $\mathbb{R}$ is the set of real numbers, and $\mathbb{I} := \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$. The ith element of the score vector S, s_i, represents the probability that the label of trace T, $l(T)$, is equal to i, $s_i = \Pr[l(T) = i]$, for $i \in \{1, \ldots, n\}$.

Once trained, the model $\mathcal{N}$ is applied to traces collected from the target device to predict the values of the intermediate variables which are represented by the labels.

The effectiveness of profiled DL-SCA relies on the model's ability to generalize from the profiling traces to the traces from the target device.

2.1.4 First- and Higher-Order Attacks

Initially introduced in [KJJ99] and formally defined in [Mes00], an nth-order side-channel attack exploits n distinct data samples from a trace, each corresponding to a different intermediate value within the cryptographic algorithm. Combining functions are used to integrate information from the n selected samples.

Higher-order attacks are particularly relevant when targeting protected implementations, especially those employing masking countermeasures [CJRR99]. For example, in an attack on a first-order Boolean masking scheme, two samples representing the Boolean shares are XORed to cancel out the random mask.

DL–based attacks are inherently higher-order, as neural networks typically operate on joint representations of multiple input features, allowing them to capture complex, non-linear relationships among these features.

2.1.5 Leakage Models

Common leakage models for SCA include the Hamming weight, Hamming distance, and identity model, each reflecting different assumptions about how data-dependent leakage arises during cryptographic computations.

Hamming weight: The *Hamming weight* of a binary vector is defined as the number of bits set to 1. This model assumes that the leakage is proportional to the number of 1s in the data being processed at the point of interest.

Hamming distance: The *Hamming distance* between two binary vectors corresponds to the number of bit positions in which they differ. This model assumes that leakage is proportional to the number of bit transitions ($0 \rightarrow 1$ and $1 \rightarrow 0$) between two consecutive states. Both types of transitions are assumed to contribute equally to the leakage.

Identity model: The *identity model* treats the actual value of the processed data (e.g., a byte or word) as the leakage itself. This model is often employed in DL–based power or EM analysis, in which neural networks are trained to learn leakage patterns associated with different classes.

2.1.6 Attack Metrics

A common metric for assessing the effectiveness of SCA is the key *rank* [CRR03] defined as the index of the correct key in the ordered list of key guesses sorted from most likely to least likely. The real key is considered successfully recovered when its rank reaches zero, or, in some cases, when it falls within a range that permits brute-force enumeration.

As a complementary evaluation metric, in this paper we additionally employ the *number of recovered subkeys*, defined as the number of subkeys for which the rank reaches zero.

2.2 Sources of Leakage in RF Signals

A comprehensive analysis of leakage sources in mixed-signal circuits integrating radio transmitters is presented in [CPM+18]. This section provides a brief summary.

Leakage Generation: In the digital domain of a mixed-signal circuit, logic operations are implemented through transistors switching between high and low voltage states, corresponding to logical 1s and 0s. The resulting fluctuations in current are inherently data-dependent. This phenomenon forms the physical basis for side-channel attacks, as changes in current can be observed by measuring the power consumption of the device, either directly through a shunt resistor or via a proxy measurement such as EM emission.

Leakage Propagation: In a mixed-signal circuit, the analog and digital parts are fabricated on a shared silicon substrate. The substrate is a weakly conducting medium, so, as illustrated in Fig. 1, a data-dependent signal induced by the switching activity of transistors ① can propagate from the digital domain to the analog domain ②. There are three mechanisms by which a signal may propagate through the substrate: capacitive, inductive, or resistive coupling. Among these, *capacitive coupling* ③ is often the most significant.

Spectrum Dispersion: From a spectral analysis perspective, any time-domain signal can be decomposed into a sum of sinusoidal components. A digital clock is a periodic square wave consisting of a fundamental frequency and an infinite number of harmonics at its multiples. If a data-dependent signal and the clock both pass through a nonlinear element such as a transistor, the transistor effectively functions as a mixer, generating sum and difference frequencies. This interaction modulates the clock with data, causing each harmonic of the clock to become a carrier for the data signal. Consequently, data-dependent spectral components can be shifted into frequency bands within the radio's baseband signal range.

Radio Transmission: Once the data signal, shifted to a certain frequency, reaches the analog domain, it mixes with the RF carrier signal generated by the voltage-controlled oscillator (VCO) in the radio module, in the same way as previously described. The resulting modulated RF carrier is then combined with the legitimate baseband signal, amplified by the power amplifier, and transmitted via the on-chip antenna. This process results in the unintentional leakage of sensitive information over the air, enabling RF-based side-channel attacks from distances within the operational range of the radio protocol.

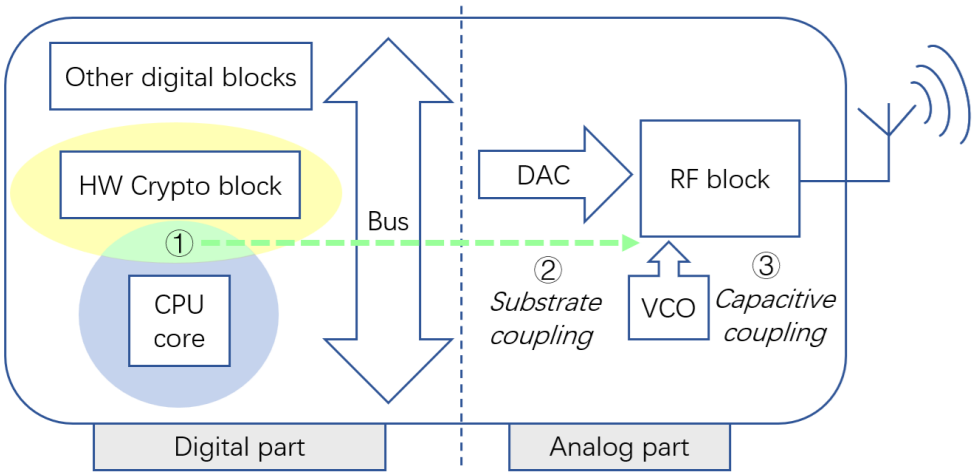

Figure 1: Illustration of leakage generation and propagation (adapted from [CPM+18]).

3 Related work

In this section, we describe foundational work on RF-based side-channel attacks, as well as recent attacks that exploit phase modulation. We also review prior work on profiled differential analysis and attacks against ASIC implementations of AES-CCM.

3.1 RF-Based Side-Channel Attacks

RF-based side-channel attacks, introduced by Camurati et al. [CPM+18], exploit unintentional information leakage modulated onto intentionally transmitted RF signals. Unlike traditional near-field EM attacks, which rely on incidental emissions, RF-based attacks leverage the RF carrier signals used in wireless communication protocols such as Bluetooth as a side channel. These attacks are generally more stealthy than near-field EM attacks, as they do not require direct access to the target device. In many cases, RF signals can be captured from several meters away, enabling remote data acquisition without alerting the device owner.

In [CPM+18], the authors demonstrated an RF-based template attack on a software implementation of AES-128 (TinyAES [tin]) operating in ECB mode on an ARM Cortex-M4 processor within the nRF52832 system-on-chip (SoC). The attack recovers the key using 52,000 RF signals with 500-sample averaging, captured at a distance of one meter. A subsequent enhancement of this attack is presented in [CFS20], where the key is recovered using 5,000 RF signals with 1,000-sample averaging, captured from a distance of 15 meters. In the same study, the authors also report their attempt to attack the AES accelerator in the nRF52832 SoC. According to their experimental resuts, the key could not be recovered from 1,000,000 RF signals with 100-sample averaging.

Wang et al. [WWDB21] presented a deep learning-based RF side-channel attack on the same AES software implementation in the nRF52832. Their approach successfully recoveres the key from fewer than 400 RF signals captured at a 15-meter distance without averaging. Training traces are acquired via coaxial cable from five profiling devices. In an earlier work [WWD20], neural networks are trained on a mixed dataset that includes traces captured at multiple distances from the profiling devices. Such an attack requires more than 1,500 RF signals with 100-sample averaging to recover the key.

Guillaume et al. [GPNS24] investigated RF-based attacks at non-harmonic frequencies. They demonstrated that key recovery remains feasible for software AES implementations, although the attack requires more RF signals due to increased signal interference. In their setting, an exhaustive search over up to 2^{32} keys is necessary for full key recovery.

Recently, Ji et al. [JDW25a, JDW25b, JDW25d] presented the first successful attack on the AES accelerator in the nRF52832 SoC operating in CBC-MAC mode. Their attack, conducted from a one-meter distance, recovers the full key from 180,000 RF signals with 10,000-sample averaging using correlation analy-

sis with up to $2^{42.8}$ enumerations. Results of reverse engineering of the AES-CCM accelerator performed by side-channel analysis with a known key are also provided. The deep learning-based RF side-channel attack on the AES-CCM accelerator presented in the follow-up work [JDW25c] recovers the full key from 45,000 RF signals with 10,000-sample averaging using up to 2^{16} enumerations.

3.2 Phase Modulation-Based Attacks

In [O4], a non-profiled attack exploiting phase modulation is presented, utilizing JTAG as a side-channel measurement method. Shortly after, Ayoub et al. [AHC+25] proposed a similar idea of exploiting phase modulation in SCA. In their attack, the full AES key can be recovered from near-field EM measurements received by an SDR through an EM probe by combining the amplitude and phase information through the addition of the absolute values of the corresponding Pearson correlation coefficients. Both papers point out that signal jitter in the time domain translates to phase shift in the frequency domain, leading to unintended leakage of secret information.

3.3 Profiled Differential Power Analysis

In [MS16], Moradi et al. proposed a profiled differential power analysis (DPA) method called *moments-correlating profiled DPA*. During the profiling stage, the dth-order statistical moments corresponding to the leakage of the target intermediate variables, such as the S-box output, are estimated based on profiling traces captured with a known key. At the attack stage, these moments are correlated with the test traces to identify the best-matching key candidate. A fundamental difference between the profiling method of [MS16] and the approach presented in this work is that we use a weighted sum of multiple samples of a trace to create a single joint score for the correlation analysis.

3.4 Attacks on ASIC Implementations of AES-CCM

Side-channel analysis of AES has been an active area of research for over two decades, with notable contributions including [OGOP04, MPO05, Kiz09, VMC19, KYSF21]. For a comprehensive overview of the field, we refer the reader to the survey in [RD20]. In this section, we specifically focus on side-channel attacks targeting ASIC implementations of AES operating in CCM mode.

O'Flynn and Chen [OC16] demonstrated a power-based side-channel attack on the hardware AES accelerator of the Atmel ATMega128RFA1 SoC. Using the same leakage model as in [Kiz09], they recover the key from 50,000 power traces captured in AES-ECB mode. They further show that key recovery is also feasible in AES-CCM mode, assuming the attacker has knowledge of the nonce. Their approach extends the method proposed by Jaffe [Jaf07], in which partial knowledge of the current round key is leveraged to infer information about the subsequent round key.

Building on this line of work, Ronen et al. [RSWO17] introduced a more advanced technique that eliminates the requirement for nonce knowledge. Their method successfully recovers the key from the hardware AES-CCM accelerator in the Atmel ATMega128RFA1 SoC used in Philips Hue smart light bulbs.

4 Threat Model

We use the same threat model as in [JDW25b].

The adversary is assumed to have physical access to the target device and the capability to capture its RF signals during the execution of the AES-CCM authenticated encryption algorithm. The adversary has control over the message and the nonce used in the encryption process. In addition, the adversary possesses a reference device identical to the target, which is used for profiling.

The primary goal of the adversary is to recover the encryption key used by the AES-CCM accelerator in the target device.

Although this work focuses on key recovery, the proposed methodology is general and can be extended to extract other forms of sensitive information.

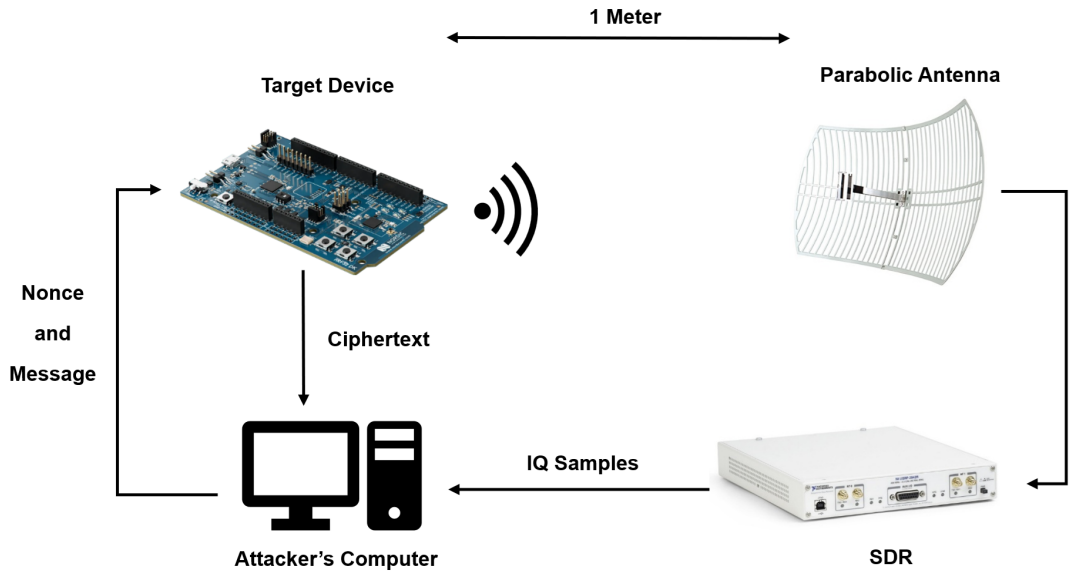

Figure 2: Experimental setup.

5 Experimental Setup

Our target device is the Nordic Semiconductor nRF52832, a general-purpose SoC consisting of an ARM Cortex-M4 processor, an AES-CCM hardware accelerator, and an nRF52 Series 2.4GHz transceiver which supports multiple protocols including BLE, NFC, ANT and a 2.4GHz proprietary protocol [nrf].

The ARM Cortex-M4 processor with floating-point unit has a 32-bit instruction set and operates at 64MHz. The CCM module, compliant with Bluetooth standards, employs an AES-128 hardware accelerator that is shared with the ECB module with a higher priority. The radio block includes a 2.4GHz radio receiver and transmitter compatible with 1 Mbps and 2 Mbps BLE modes. The CCM module and radio can be configured to work synchronously.

In our experiments, we use the USRP-2944R SDR [usr] as a receiver. To extract side-channel information from the AES accelerator, following [JDW25b], the SDR is configured to demodulate the RF signals at a center receiving frequency of $f_c = f_{\text{Bluetooth}} + n \cdot f_{\text{clock}}$, where $f_{\text{Bluetooth}} = 2.4\text{GHz}$ is the Bluetooth channel center frequency, f_{clock} is the clock frequency of the AES accelerator, and n is a positive integer. The sampling rate is set to 50 MHz.

The demodulated signals are recorded as a complex-valued representation $z(t) = I(t) + jQ(t)$, where each sample contains the in-phase (I) and quadrature (Q) parts. The instantaneous amplitude and phase components of $z(t)$ are then computed from I and Q as:

- *Amplitude :*

$$A(t) = |z(t)| = \sqrt{I(t)^2 + Q(t)^2} \tag{2}$$

- *Phase:*

$$\phi(t) = \arg(z(t)) = \tan^{-1}\left(\frac{Q(t)}{I(t)}\right)$$

- *Shifted phase:*

$$\phi_s(t) = \frac{d\phi(t)}{dt} \tag{3}$$

RF signals are captured in two settings: 1) using a coaxial cable with a 10dB attenuator connecting the target device, to simulate a noise-free environment, and 2) using a grid parabolic antenna TL-ANT2424B with 24dBi gain, to simulate a realistic attack environment. The physical setup is illustrated in Fig. 2.

5.1 AES-CCM Protocol

Counter with Cipher Block Chaining-Message Authentication Code (CCM) is an algorithm that provides both confidentiality and authenticity of data [WHF03]. It utilizes a block cipher for encryption and decryption. The CCM algorithm using AES as the block cipher, referred to as AES-CCM, forms the basis of the security architecture in the IEEE 802.11i standard [IEE04].

The CCM specification consists of two components: the Counter (CTR) mode for data encryption and the Cipher Block Chaining Message Authentication Code (CBC-MAC) for data integrity and authenticity. Unlike AES in ECB mode, where a plaintext block P is given as input to AES and the ciphertext encrypting P under the key K, $C = E(K, P)$, is produced at the output, AES-CCM encrypts counter blocks $\mathrm{Ctr}_0, \mathrm{Ctr}_1, \ldots$ consisting of a nonce N, a counter, and some additional fields. An encrypted counter block Ctr_i is called a *keystream*, $S_i = E(K, \mathrm{Ctr}_i)$. Each message block m_i is encrypted by XORing it with the corresponding keystream block, Ctr_i, starting from Ctr_1, resulting in the ciphertext $C_i = m_i \oplus S_i$, for $i \in \{1, 2, \ldots\}$. The first counter block, Ctr_0, is reserved for encrypting the MAC.

Encryption of the counter block Ctr_1 is the attack point of all attacks presented in this paper. An attacker who knows the first message block m_1 can compute the keystream S_1 from the ciphertext C_1 as $S_1 = C_1 \oplus m_1$, and subsequently recover the last round key via a side-channel attack targeting the last round of AES. An attacker who knows the nonce N used in generating S_1 can also verify the recovered key using the known plaintext-ciphertext pair (Ctr_1, S_1).

6 The Hilbert Transform

This section defines the Hilbert transform and explains reasons for employing it in SCA.

6.1 Definition and Properties

The *Hilbert transform*, named after David Hilbert (1862-1943), is a fundamental operation in signal processing and communication systems, widely employed in applications involving modulation, envelope detection, and time-frequency analysis. The Hilbert transform enables single-sideband modulation which is the most spectrally efficient ways to modulate one real-valued channel of baseband information into a passband carrier [SS10].

Conceptually, the Hilbert transform generates a version of a real-valued signal $x(t)$ that is phase-shifted by 90° (or $\pi/2$ radians). This transformed signal, denoted $\hat{x}(t)$, forms the imaginary component of the analytic signal, a complex-valued representation of the original signal $x(t)$.

Formally, the Hilbert transform $\hat{x}(t)$ of a real-valued signal $x(t)$ is defined by:

$$\hat{x}(t) = \frac{1}{\pi} \,\mathrm{P.V.} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau}\, d\tau \tag{4}$$

where P.V. denotes the *Cauchy principal value*, required due to the singularity at $\tau = t$.

The resulting *analytic signal* $z(t)$ is expressed as:

$$z(t) = x(t) + j\hat{x}(t). \tag{5}$$

In addition to the integral definition (4), the Hilbert transform of a real-valued signal $x(t)$ can also be expressed as a *convolution* with a specific distribution. This perspective is common in signal processing, where the Hilbert transform is viewed as a linear time-invariant system with a defined impulse response:

$$\hat{x}(t) = x(t) * \left(\frac{1}{\pi t}\right) \tag{6}$$

where "$*$" denotes convolution and the function $\frac{1}{\pi t}$ acts as the impulse response of the Hilbert transformer $h(t) = \frac{1}{\pi t}$. Because this function has a singularity at $t = 0$, the convolution is interpreted in the sense of the Cauchy principal value.

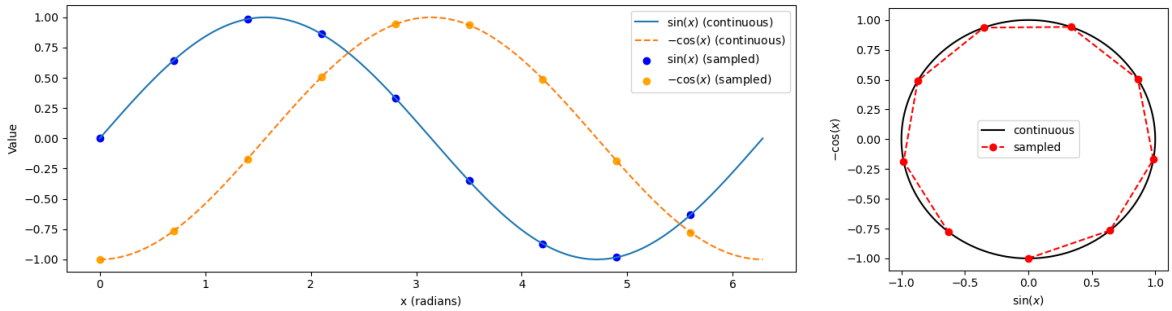

Figure 3: Signal $\sin(x)$ and its Hilbert transform, $-\cos(x)$, over the interval $0 \leq x \leq 2\pi$ (left), and the analytic signal $\sin(x) - j\cos(x)$ in the complex plane (right).

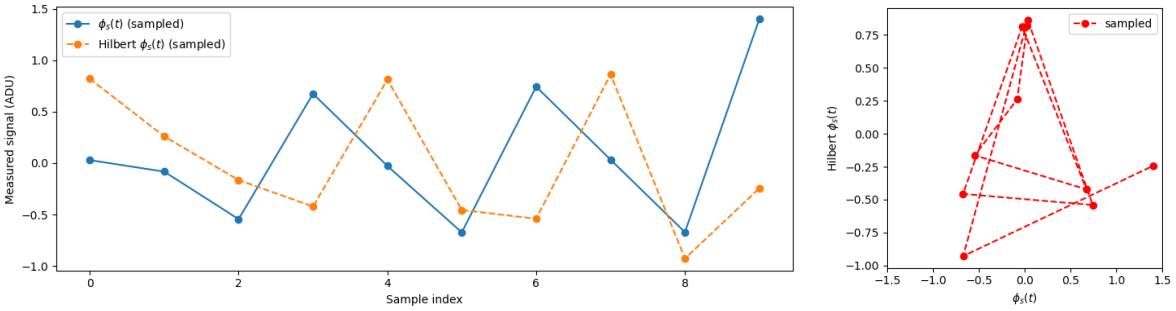

Figure 4: The shifted phase signal, $\phi_s(t)$, and its Hilbert transform, $\hat{\phi}_s(t)$, over a 10-sample interval spanning three clock cycles (left), and the analytic signal $\phi_s(x) - j\hat{\phi}_s(t)$ in the complex plane (right).

The function $\frac{1}{\pi t}$ has the following Fourier transform:

$$-j\,\mathrm{sgn}(f) = \begin{cases} -j, & f > 0 \\ 0, & f = 0 \\ j, & f < 0 \end{cases}.$$

If $x(t)$ has Fourier transform $X(f)$, then, from the convolution property of the Fourier transform, the Hilbert transform $\hat{x}(t)$ has the Fourier transform:

$$\hat{X}(f) = -j\,\mathrm{sgn}(f) \cdot X(f).$$

This representation clearly shows that the Hilbert transform is a filtering operation that rotates the phase of each frequency component by $\pm\pi/2$ without affecting the magnitude of $X(f)$.

6.2 Employing the Hilbert Transform in SCA

We start by considering the basic case of a pure sine wave, $\sin(x)$, and its Hilbert transform, $-\cos(x)$. Fig. 3 (left) shows the plots of these signals in the interval $0 \leq x \leq 2\pi$. By definition (6), $-\cos(x)$ is phase-shifted from $\sin(x)$ by 90°, or $\pi/2$. Fig. 3 (right) shows a parametric plot illustrating the trajectory of the corresponding analytic signal $\sin(x) - j\cos(x)$ in the complex plane (known as *Lissajous figure*). As expected, the trajectory is the unit circle enclosing the point of origin.

We next analyze a complex-valued signal $z(t) = I(t) + jQ(t)$ captured from the target device during the execution of AES-CCM using an SDR. Since the AES-CCM accelerator is a synchronous digital circuit, its activity is driven by a clock signal. As mentioned earlier, a clock is a periodic square wave consisting of a fundamental frequency and an infinite number of harmonics at its multiples. Each harmonic may

carry information about logic operations in the AES-CCM accelerator due to the spectrum dispersion effect explained in Section 2.2.

From $z(t)$, we derive the amplitude, $A(t)$, and shifted phase, $\phi_s(t)$, using equations (2) and (3), respectively, and compute their corresponding Hilbert transforms, $\hat{A}(t)$, and $\hat{\phi}_s(t)$. Below, we illustrate the case of the shifted phase, as it is the most representative of the two.

Fig. 4 (left) shows the plots of the shifted phase $\phi_s(t)$ and its Hilbert transform $\hat{\phi}_s(t)$ signals, over a 10-sample interval. This interval is used as input to all attacks presented in this paper. It is centered around the last round of AES encryption, which is our chosen attack point. Fig. 4 (right) also shows the trajectory of the corresponding analytic signal, $z(t) = \phi_s(t) - j\hat{\phi}_s(t)$, in the complex plane. The trajectory contains three distinct triangular loops, implying that the 10-sample interval covers three clock cycles. The shapes of the loops suggest that $\phi_s(t)$ varies in a periodic manner.

Since the complex-valued signal $z(t)$ is acquired at a sampling rate of 50 MHz, we can deduce that the underlying AES clock operates at $\lfloor 50/3 \rfloor = 16$ MHz. This estimation is consistent with the result reported in [JDW25b], where the same clock frequency is inferred through SCA with a known key.

By dividing the sampling frequency by the estimated clock frequency, we can determine that each clock cycle contains $50/16 = 3.125$ samples. Consequently, the 90° phase shift of the first (dominant) harmonic of the clock signal corresponds to a temporal shift of $3.125/4 = 0.781 \approx 1$ sample. This aligns with the one-sample phase offsets observed between some peaks of $\phi_s(t)$ and $\hat{\phi}_s(t)$ in Fig. 4 (left), e.g. for the peak of $\phi_s(t)$ at $t = 3$ and the peak of $\hat{\phi}_s(t)$ at $t = 4$, as well as for the peak of $\phi_s(t)$ at $t = 6$ and the peak of $\hat{\phi}_s(t)$ at $t = 7$. We leverage this one-sample temporal offset to construct an effective fused score that combines multiple correlation coefficients into a single metric for correlation analysis.

7 Problem Setup

Previous work in CPA has shown that analyzing information leakage is more effective when using not only raw time-domain signals but also alternative signal representations or derived signal components. They may reveal complementary leakage that is not readily visible in the original signal. For instance, transforming time-domain signals to the frequency domain has been shown to enhance leakage detectability [Tiu05]. Similarly, using both amplitude and shifted phase components of a complex-valued signal can boost CPA results [AHC+25].

In our case, we use not only amplitude, $A(t)$, and shifted phase, $\phi_s(t)$, components of a complex-valued time-domain signal, but also their Hilbert transforms, $\hat{A}(t)$ and $\hat{\phi}_s(t)$. We correlate each of these four signal components against two hypothetical leakage models: the Hamming weight of an intermediate computation, h_{HW}, and the Hamming distance between two intermediate computations, h_{HD}. The corresponding eight Pearson correlation coefficients are:

$$r(x, h) = \mathrm{corr}(x, h),$$

for $x \in \{A(t), \hat{A}(t), \phi_s(t), \hat{\phi}_s(t)\}$ and $h \in \{h_{\mathrm{HW}}, h_{\mathrm{HD}}\}$. We seek a combining function that fuses these coefficients into a single score for CPA.

7.1 Candidate Fusion Scores

We consider the following two candidates for constructing a joint score R that combines the correlation coefficients $r(x)$ and $r(y)$. These candidates have proven useful for SCA in the past [Mey12, AHC+25].

Multiplicative Score

$$R_{\mathrm{prod}} = r(x) \cdot r(y).$$

The multiplicative score emphasizes instances where both components simultaneously correlate with the leakage model. Such a fusion method is used in [Mey12].

Additive Score

$$R_{\text{sum}} = |r(x)| + |r(y)|. \tag{7}$$

The additive score accumulates evidence of leakage from either component. It is tolerant to asymmetric leakage where only one of the two components strongly correlates with the leakage model at a given time. Such a fusion method is used in [AHC+25].

The optimal fusion strategy is dependent on the inherent characteristics of the leakage in the target implementation. When leakage occurs simultaneously in both components, a multiplicative fusion score offers effective detection with enhanced suppression of false positives. Conversely, if leakage appears in only one component at a time, additive fusion may be more appropriate. To determine the most suitable fusion strategy, we therefore rely on empirical evaluation.

7.2 Fusion Score Selection

We choose an optimal fusion strategy empirically by analyzing a set of complex-valued signals $z(t) = I(t) + jQ(t)$ extracted from RF signals captured from profiling devices as described in Section 5.

Fig. 5 shows Pearson correlation coefficients computed for the components $A(t)$, $\hat{A}(t)$, $\phi_s(t)$, and $\hat{\phi}_s(t)$ of the signal $z(t)$ using the state Hamming weight and the state Hamming distance leakage models, respectively (the leakage models are described in more details in Section 9.1). The plots are derived based on 332,000 RF signals captured from three different profiling devices during the execution of the AES-CCM encryption for known keys, nonces and messages selected at random. Each encryption is repeated 10,000 times to reduce noise. For each measurement, the components $A(t)$ and $\phi_s(t)$ are first derived from $z(t)$ using equations (2) and (3) and then averaged.

An important conclusion we can draw from Fig. 5 is that a one-sample temporal offset between $A(t)$ and $\hat{A}(t)$, and $\phi_s(t)$ and $\hat{\phi}_s(t)$ can be utilized by combining $r(A, h)$ at sample j with $r(\hat{A}, h)$ at sample $j + 1$, as well as $r(\phi_s, h)$ at sample j with $r(\hat{\phi}_s, h)$ at sample $j + 1$.

To determine whether the additive or multiplicative score is the more suitable fusion strategy, we evaluate both empirically. Fig. 6 presents the results of combining the correlation coefficients of all eight components using these fusion scores. The multiplicative score is scaled as $\times 10^{10}$. All the Hilbert components are shifted by one sample; that is, sample j of $x(t)$ is fused with sample $j + 1$ of $\hat{x}(t)$. It is clear from Fig. 6 that the multiplicative score is worse than the additive, as it cancels out the peak at sample 4. We therefore select the additive score type for the subsequent analysis.

8 Weighted Sum Correlation Analysis Method

In this section, we introduce a novel SCA method referred to as the *weighted sum correlation analysis*. This method is a higher-order profiled attack, comprising a profiling and an attack phases, akin to template attacks and deep learning-based approaches. However, it differs significantly in its profiling strategy.

8.1 Profiling stage

Instead of constructing a neural network model or generating a template for each possible value of the secret, e.g., the 256 possible values of an 8-bit subkey, the weighted sum CPA derives a set of *weights* for each sample that prioritize samples with higher average leakage, irrespective of the secret value. These weights reflect the importance of each sample in terms of its statistical relevance to the chosen leakage model. Our experimental results show that such an approach effectively reduces the impact of non-informative samples, thereby improving robustness to noise. Furthermore, it enhances cross-device portability when profiling and attack phases are performed on different devices (as in a real attack).

The weights are obtained by computing the Pearson correlation coefficients for each signal component, each leakage model, and each row of the AES state. Recall that the AES state is treated as the following

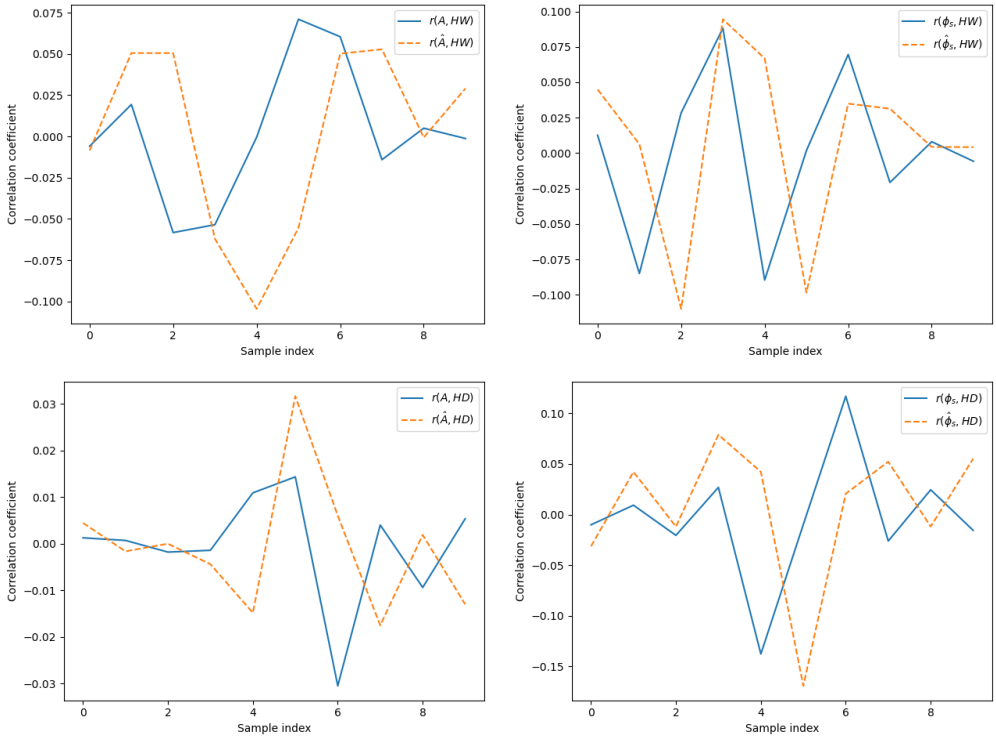

Figure 5: Pearson correlation coefficients computed for $A(t)$, $\hat{A}(t)$, $\phi_s(t)$, and $\hat{\phi}_s(t)$ using the state Hamming weight and the state Hamming distance leakage models.

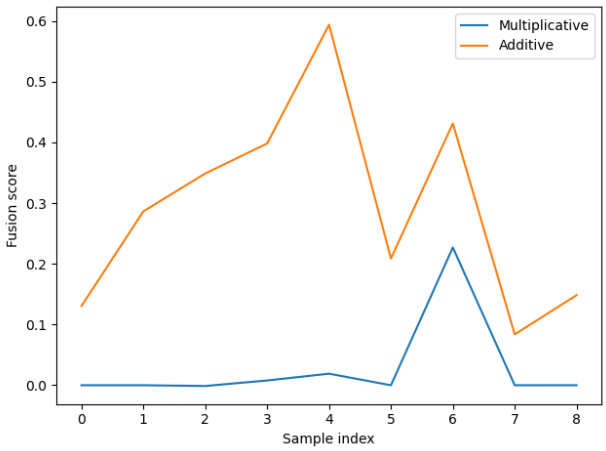

Figure 6: Comparison of additive and multiplicative fusion scores.

4×4 matrix of bytes b_i, for $i \in \{0, 1, \ldots, 15\}$:

$$\text{AES state} = \begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}.$$

Let $\mathbf{x}_j = [x_{1,j}, x_{2,j}, \ldots, x_{n,j}]^\top$ be the vector of values of the signal component $x \in \{A(t), \hat{A}(t), \phi_s(t), \hat{\phi}_s(t)\}$ at sample j across n traces, and let $\mathbf{h_s} = [h_{1,s}, h_{2,s} \ldots, h_{n,s}]^\top$ be the vector of corresponding values of leakage model $h \in \{h_{\text{HW}}, h_{\text{HD}}\}$ for the subkey s for those signal values, where $s \in \{0, 1, \ldots, 15\}$ for AES-128. Then, according to the eq. (1), the Pearson correlation coefficient at sample j for the signal component x and subkey s under the leakage model h is given by:

$$r_{j,s}(x,h) = \frac{\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)(h_{i,s} - \bar{h}_s)}{\sqrt{\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)^2}\sqrt{\sum_{i=1}^{n}(h_{i,s} - \bar{h}_s)^2}}, \tag{8}$$

where $\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}$ and $\bar{h}_s = \frac{1}{n}\sum_{i=1}^{n} h_{i,s}$ are the means of the signal values at sample j and leakage hypothesis values for subkey s, respectively.

The weight $w_{j,s}(x,h)$ at sample j for the signal representation x and subkey s under the leakage model h is equal to the sum of four correlation coefficients of the subkeys in the same row of the AES state as subkey s:

$$w_{j,s}(x,h) = r_{j,s}(x,h) + r_{j,s\boxplus 4}(x,h) + r_{j,s\boxplus 8}(x,h) + r_{j,s\boxplus 12}(x,h), \tag{9}$$

where "$\boxplus$" is addition modulo 16. Note that the weights can be both positive or negative, since we are not taking the absolute value of the correlation coefficients.

We define weights across each row of the AES state rather than at the level of a single byte, as a row-wise construction ensures that even weakly leaking bytes receive meaningful weights. Our experiments show that, in the target AES-CCM implementation, byte leakage is highly non-uniform. We chose not to define weights across the entire state because the leakage profiles of different rows vary significantly, especially for the first row whose bytes b_0, b_4, b_8, b_{12} are not shifted by the ShiftRows operation.

8.2 Attack stage

Each of the four signal representations $A(t), \hat{A}(t), \phi_s(t), \hat{\phi}_s(t)$ is subjected to correlation analysis using the byte Hamming weight h_{HW}, and the byte Hamming distance, h_{HD}, leakage models, resulting in a set of eight matrices across all key hypotheses and samples. For each of the 16 subkeys, the Pearson correlation coefficients for these matrices are individually computed using the eq. (8).

The presented method aggregates the correlation coefficients by computing their weighted sum based on the set of weights derived at the profiling stage. Let $r_{j,s,k}(x,h)$ be the Pearson correlation coefficient at time j for guess k of subkey s and signal representation x under the leakage model h, and $w_{j,s}(x,h)$ is the corresponding weight. Then, the composite correlation score for guess k of subkey s given by:

$$R_{s,k} = \sum_j R_{j,s,k} \tag{10}$$

where

$$R_{j,s,k} = \sum_{h \in \{h_{\text{HW}}, h_{\text{HD}}\}} \left(\sum_{x \in \{A(t), \phi_s(t)\}} w_{j,s}(x,h) \cdot r_{j,s,k}(x,h) + \sum_{x \in \{\hat{A}(t), \hat{\phi}_s(t)\}} w_{j,s}(x,h) \cdot r_{j+1,s,k}(x,h) \right). \tag{11}$$

The most likely subkey is identified by selecting the subkey guess k whose corresponding score vector $R_{s,k}$ has the highest maximum value:

$$k^* = \arg\max_k R_{s,k}. \tag{12}$$

The process is repeated independently for each subkey $s \in \{0, 1, \ldots, 15\}$.

Note that, while in eq. (11) $R_{j,s,k}$ is computed additively, we do not take the absolute values of the terms like in the additive score R_{sum} in eq. (7). This is intentional, as it allows us to identify cases where the sign of $r_{j,s,k}(x,h)$ differs from that of the weight $w_{j,s}(x,h)$. If the signs differ, the additive score $R_{j,s,k}$ decreases. Since we expect the sign of the weight computed row-wise to be correct, this strategy effectively discriminates wrong subkey guesses.

9 Experimental Results

In this section, we evaluate the proposed weighted sum correlation analysis method and compare it to the CPA approaches presented in CHES'25 and RWC'25, and to DL-based SCA.

9.1 Profiling details

At the profiling stage, we collect a total of 332,000 RF signals from three different profiling devices connected via a coaxial cable. The RF signals are collected during the execution of AES-CCM encryption by the AES accelerator for known keys, nonces and messages selected at random and demodulated into the complex-valued signals $z(t) = I(t) + jQ(t)$ as described in Section 5. To reduce noise, each encryption is repeated $N = 10,000$ times. For each repetition, the amplitude $A(t)$ and the shifted phase $\phi_s(t)$ are extracted from $z(t)$ using equations (2) and (3). Once all 10,000 repetitions are performed, the resulting $A(t)$ and $\phi_s(t)$ components are averaged and saved. In this way, according to the square root scaling law [Mar07], the signal-to-noise ratio is increased by a factor of $\sqrt{10,000} = 100$. It is worth mentioning that averaging the complex-valued signals $z(t)$ directly, prior to extracting $A(t)$ and $\phi_s(t)$, results in a loss of phase information due to the varying initial phases of individual $z(t)$ signals.

Then, we then derive the Hilbert transforms of $A(t)$ and $\phi_s(t)$, $\hat{A}(t)$ and $\hat{\phi}_s(t)$. For each of these four signal components $x \in \{A(t), \hat{A}(t), \phi_s(t), \hat{\phi}_s(t)\}$, we construct two leakage models, $h = \{h_{\text{HW}}, h_{\text{HD}}\}$, for each subkey $s \in \{0, 1, \ldots, 15\}$.

The h_{HW} leakage model for subkey s is computed as the Hamming weight of the last round's S-Box input, $\text{HW}(state_9)$, where $state_9$ is given by:

$$state_9 = \text{SBox}^{-1}(S_1[s] \oplus RK_{10}[s]),$$

where SBox^{-1} denotes the inverse of the S-Box operation and $S_1[s]$ and $RK_{10}[s]$ are the sth bytes of the keystream S_1 and the last round key RK_{10}, respectively.

The h_{HD} leakage model for subkey s is computed as the Hamming distance between $state_9$ and the sth byte of $\text{ShiftRows}^{-1}(S_1)$:

$$state_{10} = \text{ShiftRows}^{-1}(S_1)[s],$$

where ShiftRows^{-1} is the inverse of the ShiftRows operation. This is equivalent to $\text{HW}(state_9 \oplus state_{10})$. This leakage model is known as the *last round state difference*.

Finally, the weights $w_{j,s}(x,h)$ at sample j for the signal representation x and subkey s under the leakage model h are computed using the eq. (9) as the sum of four correlation coefficients of the subkeys in the same row of the AES state as subkey s.

9.2 Attack details

At the attack stage, we perform AES-CCM encryptions on the target device using a set of known messages and nonces selected at random and a fixed unknown secret key. The RF signals are collected under two different settings:

1. In physical proximity to the target device using a coaxial cable, and
2. At a one-meter distance from the target device using an antenna, in an office environment.

For both settings, we capture traces from two nRF52832 devices (different from those used for profiling), referred to as device 1 and device 2. Each device uses a different fixed key, referred to as key 1 and key 2, selected at random.

As in [JDW25a, JDW25b], the attack is performed on the encryption of the counter block Ctr_1. Recall that, in the CCM mode, the keystream S_1, representing the encrypted Ctr_1, is used as a pad to encrypt the first message block, m_1, resulting in the ciphertext $C_1 = m_1 \oplus S_1$. Hence, an attacker who uses the device under attack for the encryption of a known message block m_1 can determine S_1 as $S_1 = C_1 \oplus m_1$.

Using known keystreams S_1, for each candidate $k \in \{0, 1, \ldots, 255\}$ of a given subkey $s \in \{0, 1, \ldots, 15\}$, each leakage model $h \in \{h_{\mathrm{HW}}, h_{\mathrm{HD}}\}$, and each signal component $x \in \{A(t), \hat{A}(t), \phi_s(t), \hat{\phi}_s(t)\}$, we compute the Pearson correlation coefficient $r_{j,s,k}(x, h)$ at sample j using Welford's approximation [Wel62]. This approximation can be computed much more efficiently than a naive approach that directly follows eq. (1) without affecting the results. The speedup is essential to enable us to repeat each attack 100 times. The weights $w_{j,s}(x, h)$ derived at the profiling stage are then used to combine the computed correlation coefficients into the composite correlation score $R_{s,k}$ according to eq. (10). Finally, the most likely candidate for subkey s is identified as $\arg\max_k R_{s,k}$.

The process is repeated for all subkeys s to recover the full last round key, RK_{10}. Then, the original key K is computed from RK_{10} using the reverse key expansion algorithm and verified against a known plaintext-ciphertext pair (Ctr_1, S_1). If K is not correct, the brute-force enumeration method described in the next section is applied as a post-processing step.

9.3 Benchmarking against the state-of-art CPA and DL methods

We evaluated the weighted sum correlation analysis method and compared it to the correlation analysis methods presented at RWC'25 [JDW25a, JDW25b] and CHES'25 [AHC+25], as well as to a DL-based attack. All attacks are applied on the same sets of attack traces, under the same conditions.

The RWC'25 correlation analysis method uses only the amplitude component of a complex-valued signal $z(t)$ and the same Hamming weight leakage model h_{HW} as in the weighted sum method. The original attack in [JDW25a] is performed on RF signals averaged over 10,000 samples. We carry out all attacks on RF signals averaged over 1,000 samples. In Section 9.5, we show that this reduces the overall capture time (and thus access time to the device under attack) by a factor of five since the number of traces required for an attack increases only by a factor of two.

The CHES'25 correlation analysis method utilizes both the amplitude and shifted phase components of $z(t)$. The original CHES'25 attack is performed on a software implementation of AES-128 in ECB mode running on the ARM Cortex-M4 processor of the nRF52832. It uses near-field EM emissions as a side channel and the Hamming weight of the first round S-Box output as the leakage model. We replicated the CHES'25 attack on the AES accelerator in the nRF52832 using RF signals as the side channel to maintain consistency with other attacks. The leakage model is modified to the Hamming weight of the last round S-Box input as no observable leakage is detected at the first round S-Box output.

The DL-based attack is implemented similarly to the DL-based attack of [JDW25c], using an MLP neural network with three dense layers as a classifier. For a fair comparison with correlation analysis-based attacks, which use four signal components $A(t), \hat{A}(t), \phi_s(t), \hat{\phi}_s(t)$ and two leakage models h_{HW}, h_{HD}, the same four signal components are given as input to the neural network as concatenated intervals and both the value of the S-Box input in the last round and the value of the last round state difference are used as output (i.e. the output size is 512).

Table 1 summarizes the attacks results. It lists the average number of traces required to recover 12 key bytes computed over 100 attacks, each carried out on a different subset of traces. The remaining four key bytes can be brute forced by enumerating $\binom{16}{4} \times 256^4 \approx 2^{42.8}$ possible guesses and verifying each on a known plaintext-ciphertext pair. This can be done in 13 minutes on a consumer-grade GPU performing 9.998×10^9 encryptions per second [LSSH22].

We can see that the weighted sum method achieves at least a threefold reduction in the average number of traces required for key recovery compared to other correlation analysis methods. It also outperforms the DL-based attack. Among the four methods, it is the only one capable of recovering 12 key bytes from antenna-captured traces within the available trace budget. These results indicate that the weighted sum method has greater robustness to noise and inter-device differences arising from manufacturing process variation.

Fig. 7 presents the average subkey rank and the number of recovered subkeys as functions of the number of traces for the most difficult case of antenna-captured traces from device 2. The plots clearly show that the weighted sum method significantly accelerates rank convergence. Furthermore, we can see that the difficulty

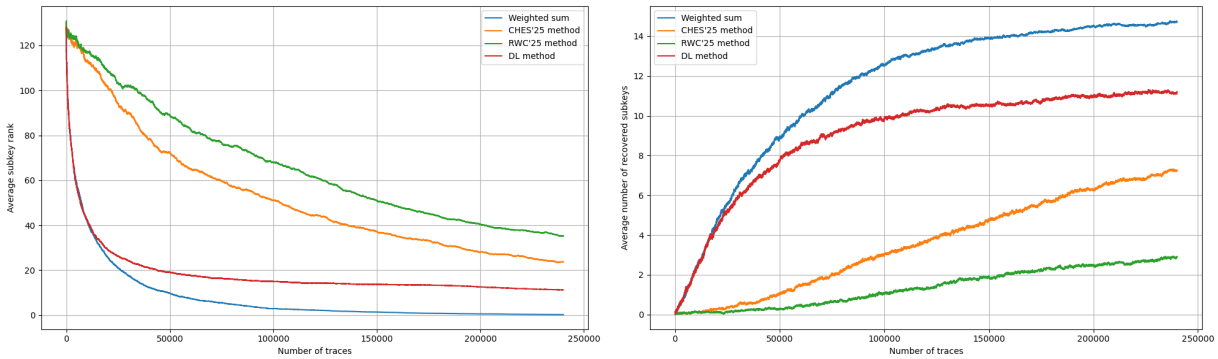

Figure 7: Comparison of average subkey rank (left) and number of key bytes recovered (right) by the weighted sum, CHES'25, RWC'25, and DL methods from RF signals captured via antenna from device 2 with 1,000-sample averaging (over 100 attacks).

Table 1: Average number of traces required to recover 12 key bytes from RF signals with 1,000 averaged samples (over 100 attacks). The format $N \times x$ represents the number of traces N and the x-fold increase relative to the weighted sum method. The dash "–" indicates failure to recover 12 key bytes within the available number of traces.

Attack Method	Device 1 Key 1			Device 2 Key 2		
	Cable (×Increase)		Antenna	Cable (×Increase)		Antenna
RWC'25	84,803	×4.07	–	–		–
CHES'25	73,072	×3.51	–	113,915	×4.78	–
DL	21,603	×1.04	–	33,831	×1.42	–
Weighted sum	**20,799**	**×1.00**	**76,346**	**23,802**	**×1.00**	**88,356**

of recovering individual subkeys in this AES implementation varies significantly. On average, it takes 88,356 traces to recover 12 subkeys and 156,685 traces to recover 14 subkeys using the weighted sum approach. This suggests that, beyond a certain point, brute-forcing the remaining subkeys is more efficient than continuing side-channel data collection. Capturing 1,000 RF signals with 1,000 repetitions takes 0.25 hours; therefore, recovering 14 subkeys instead of 12 subkeys requires $((156,685 - 88,356)/1,000) \times 0.25 = 17.1$ additional hours of capture time.

9.4 Evaluating new ideas

Since the weighted sum method employs both known techniques and new ideas, it is interesting to evaluate how much each new idea contributes to its effectiveness. Table 2 presents a summary of the evaluation across five cases. In all of them, both amplitude and shifted phase signal components, $A(t)$ and $\phi_s(t)$, as well as both leakage models, h_{HW} and h_{HD}, are used in the attack.

1. Profiled higher-order weighted sum correlation analysis without the Hilbert transform, i.e. only $A(t)$ and $\phi_s(t)$ signal components are used.

2. Profiled first-order correlation analysis with the Hilbert transform. The weights derived at the profiling stage are used to compute each composite correlation score $R_{j,s,k}$ at time j for guess k of subkey s as (11) and then the most likely candidate for the subkey s is identified as $\arg\max_{k,j} R_{j,s,k}$ (as in the classical CPA).

3. As (2) but without the Hilbert transform.

4. Non-profiled first-order correlation analysis with the Hilbert transform. Each $R_{j,s,k}$ is computed as (11)

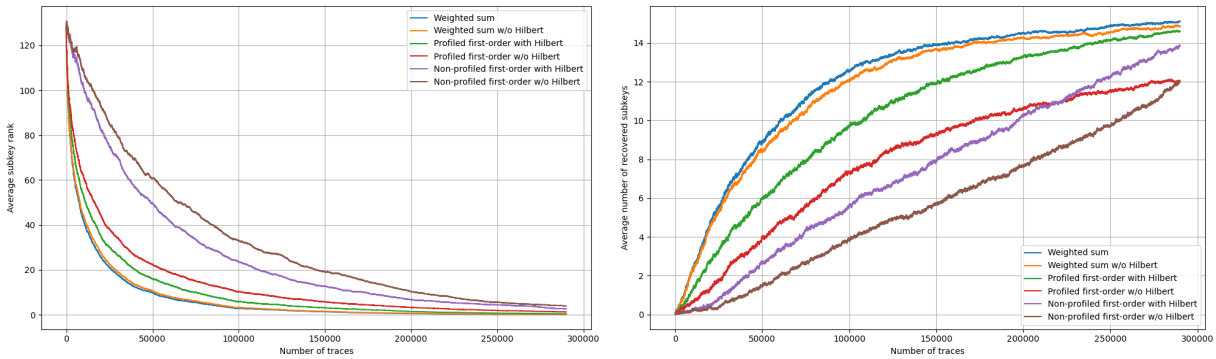

Figure 8: Comparison of average subkey rank (left) and number of key bytes recovered (right) by various modifications of the weighted sum method from RF signals captured via antenna from device 2 with 1,000-sample averaging (over 100 attacks).

Table 2: Average number of traces required to recover 12 key bytes from RF signals with 1,000 averaged samples captured from device 2 by various modifications of the weighted sum method (over 100 attacks). The format $N \times x$ represents the number of traces N and the x-fold increase relative to the 88,356 traces of the original method.

Attack Method	Device 2 Key 2	
	Antenna	(×Increase)
Weighted sum without Hilbert	97,024	×1.10
Profiled first-order with Hilbert	150,718	×1.71
Profiled first-order without Hilbert	272,937	×3.09
Non-profiled first-order with Hilbert	241,960	×2.74
Non-profiled first-order without Hilbert	289,121	×3.27

with all weights $w_{j,s}$ set to 1 and then the most likely candidate for the subkey s is identified as $\arg\max_{k,j} R_{s,j,k}$.

5. As (4) but without the Hilbert transform.

We can see from Table 2 and Fig. 8 that the Hilbert transform yields a substantial improvement in the performance of first-order attack variants, both in profiled and non-profiled settings. This may be useful in attack scenarios where a profiling device is not available. While the Hilbert transform also provides a 10% gain in the higher-order profiled attack, the improvement is less pronounced since the baseline is already highly optimized.

We can also see that all first-order weighted sum variants outperform the CHES'25 and RWC'25 methods. The latter fail to recover 12 key bytes from antenna-captured RF signals from device 2, whereas each of the former methods achieves this successfully.

9.5 Verifying the impact of averaged samples

The use of different signal acquisition strategies for profiling and attack stages may raise concerns regarding their potential impact on the performance of DL models. Specifically, we employ cable-captured RF signals with 10,000 averaged samples for profiling, while RF signals for attacks are captured with 1,000 averaged samples via both cable and antenna. In this section, we demonstrate that these differences do not negatively affect DL model performance.

Our profiling strategy is motivated by prior work [WWDB21] showing that training DL models on high-quality traces improves performance in RF-based side-channel attacks. The attack on a software AES implementation running on an ARM Cortex-M4 processor in the nRF52832 chip presented in [WWDB21]

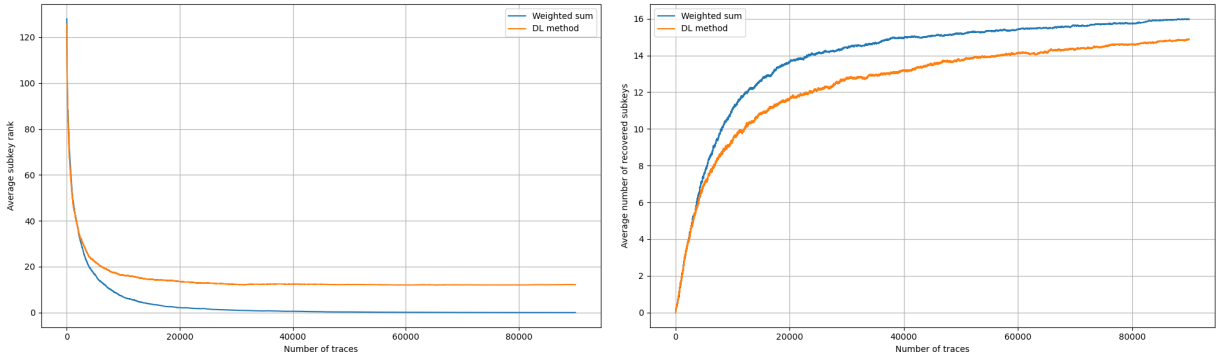

Figure 9: Comparison of average subkey rank (left) and number of key bytes (right) recovered by the weighted sum and DL methods methods from RF signals captured via cable from device 2 with 10,000-sample averaging (over 100 attacks).

successfully recovers the key from less than 400 RF signals captured via an antenna positioned 15 meters from the target device without applying averaging. In that attack, DL models are trained on "clean" cable-captured RF-signals with 100-sample averaging. In contrast, an earlier attack on the same AES implementation [WWD20] using DL models trained on a mixed dataset of RF signals with 100-sample averaging captured via antenna at various distances to the target (1, 2, 4, 8 meters) and via cable, requires more than 1,500 RF signals with 100-sample averaging to recover the key in the same attack setting as in [WWD20].

To assess whether the differences in profiling and attack conditions affects DL model performance in our setup, we re-captured a test set under the same conditions used for training, namely via cable with 10,000 averaged samples. The results shown in Fig. 9 confirm that the weighted sum correlation analysis continues to outperform the DL-based approach: on average, 12,196 traces are required to recover 12 key bytes using the former, compared to 22,798 traces using the latter.

10 Discussion

During the last decade, we observed the birth and rise of deep learning-based SCA [MPP16]. DL-based attacks on commercial devices reducing the number of attack traces by three orders of magnitude compared to CPA have been demonstrated, e.g. [BFD21]. However, that decade can be regarded as a "dark age" for academic research in the field because, due to the black-box nature of DL, the fundamental knowledge hardly increased. There is a significant gap between the empirical success of DL-based attacks and our theoretical understanding of the underlying mechanisms.

Deep learning has proven to be a powerful tool in SCA, but its application requires careful consideration. While optimization of DL architectures and hyperparameters may yield reductions in the number of required attack traces, studies focusing primarily on such improvements often make limited contributions to advancing the state of the art. Given the vast design space of neural networks, it is typically possible to find a DL model that performs marginally better under specific conditions. However, progress in SCA is ultimately driven by a deeper understanding of the fundamental attack principles.

In this paper we present a correlation analysis-based attack method which outperforms DL-based approach while remaining computationally lightweight and interpretable. We believe that we barely scratch the surface of its potential. Future work may consider more complex weighted combining functions, potentially multi-level. For instance, in eq. (11), the first level may combine $r_{j,s,k}(x,h)$ of each signal representation x with $r_{j+1,s,k}(x,h)$ of its Hilbert counterpart using one operation. On the second level, the resulting terms for the amplitude and phase signal components may be combined using another operation. Finally, on the third level, the terms for the leakage models h_{HW} and h_{HD} may be joined by yet another operation.

Another possible direction is to evaluate other fusion scores, e.g. ℓ_2 norm (Euclidean distance), defined

by

$$R_{\ell_2} = \sqrt{r(x)^2 + r(y)^2}$$

and ℓ_∞ norm (max norm) defined by

$$R_{\ell_\infty} = \max\left(|r(x)|,\ |r(y)|\right).$$

The ℓ_2 norm measures the magnitude of the correlation vector $\vec{r} = (r(x), r(y))$ in the two-dimensional space, balancing the contributions of both x and y components. The ℓ_∞ norm reflects the strongest individual correlation among the components, focusing on the dominant leakage at each time point.

Yet another possibility is to perform correlation analysis directly on the complex-valued signals by utilizing rotational and reflectional complex correlations instead of the Pearson correlation coefficient. *Rotational* correlation describes how data is transformed under rotation. Hence, it allows to detect relationships which are independent of global phase rotation, i.e., if two signals differ only by a constant phase shift, their rotational correlation remains high. *Reflectional* correlation, on the other hand, reflect how data is transformed under mirroring or flipping. It is effective for detecting similarity between signals that may be complex conjugates of one another, e.g., that appear as mirror images.

11 Conclusion

In this work, we introduced the weighted sum correlation analysis, a profiled higher-order side-channel attack that enhances classical CPA through sample weighting and signal component fusion. By quantifying the contribution of individual time-domain samples to leakage during the profiling phase, the presented method improves attack effectiveness without relying on deep learning models.

We further demonstrated the value of incorporating signal diversity in the form of the original signal and its Hilbert transform, and reconfirmed the advantage of employing both the amplitude and phase signal components as well as multiple leakage models. These forms of redundancy are fused using the weighted sum approach, which significantly outperforms prior state-of-the-art CPA-based attacks, achieving at least a threefold reduction in the number of required attack traces. It also outperforms a deep learning-based attack while remaining computationally lightweight and interpretable. We hope this contribution will encourage a renewed focus on interpretable techniques in SCA, ending the "dark age" of deep learning dominance.

References

[AARR03] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM side-channel(s). In *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 29–45. Springer Berlin Heidelberg, 2003.

[AHC+25] Pierre Ayoub, Aurélien Hernandez, Romain Cayre, Aurélien Francillon, and Clémentine Maurice. PhaseSCA: Exploiting phase-modulated emanations in side channels. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2025(1):392–419, 2025.

[ARR03] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi-channel attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2003*, pages 2–16. Springer Berlin Heidelberg, 2003.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems - CHES 2004*, pages 16–29. Springer Berlin Heidelberg, 2004.

[BFD21] Martin Brisfors, Sebastian Forsmark, and Elena Dubrova. How deep learning helps compromising USIM. In *Smart Card Research and Advanced Applications*, pages 135–150. Springer International Publishing, 2021.

[CFS20] 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*, 2020(3):358–401, 2020.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology - CRYPTO '99*, volume 1666, pages 398–412. Springer, 1999.

[CPM+18] 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*, page 163–177. ACM, 2018.

[CRR03] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In Burton S. Kaliski, çetin K. Koç, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2002*, pages 13–28. Springer Berlin Heidelberg, 2003.

[Dub13] Elena Dubrova. *Fault-tolerant design*. Springer, 2013.

[GMGH19] Christophe Genevey-Metat, Benoît Gérard, and Annelie Heuser. Combining sources of side-channel information. In *C&ESAR 2019*, 2019.

[GPNS24] Jeremy Guillaume, Maxime Pelcat, Amor Nafkha, and Ruben Salvador. Attacking at non-harmonic frequencies in screaming-channel attacks. In *Smart Card Research and Advanced Applications*, pages 87–106. Springer Nature Switzerland, 2024.

[HWW22] Fanliang Hu, Huanyu Wang, and Junnian Wang. Multi-leak deep-learning side-channel analysis. *IEEE Access*, 10:22610–22621, 2022.

[IEE04] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Medium access control (MAC) security enhancements, ANSI. *IEEE Std*, 802, 2004.

[Jaf07] Josh Jaffe. A first-order DPA attack against AES in counter mode with unknown initial counter. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 1–13. Springer Berlin Heidelberg, 2007.

[JDW25a] Yanning Ji, Elena Dubrova, and Ruize Wang. Is your Bluetooth chip leaking secrets via RF signals?, 2025. `https://www.youtube.com/watch?v=qIs0S-DNhCE&ab_channel=IACR`.

[JDW25b] Yanning Ji, Elena Dubrova, and Ruize Wang. Is your Bluetooth chip leaking secrets via RF signals? Cryptology ePrint Archive, Paper 2025/559, 2025.

[JDW25c] Yanning Ji, Elena Dubrova, and Ruize Wang. Is your chip leaking secrets via RF signals? In *2025 IEEE 55th International Symposium on Multiple-Valued Logic (ISMVL'2025)*, pages 141–146. IEEE Computer Society, 2025.

[JDW25d] Yanning Ji, Elena Dubrova, and Ruize Wang. Screaming channels revisited: Encryption key recovery from AES-CCM accelerator. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS'2025)*, pages 1–5, 2025.

[Kiz09] Ilya Kizhvatov. Side channel analysis of AVR XMEGA crypto engine. In *Proceedings of the 4th Workshop on Embedded Systems Security*, page 3. ACM, 2009.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology – CRYPTO' 99*, pages 388–397. Springer Berlin Heidelberg, 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology — CRYPTO '96*, pages 104–113. Springer Berlin Heidelberg, 1996.

[KYSF21] Takaya Kubota, Kota Yoshida, Mitsuru Shiozaki, and Takeshi Fujino. Deep learning side-channel attack against hardware implementations of AES. *Microprocessors and Microsystems*, 87:103383, 2021.

[LSSH22] Wai-Kong Lee, Hwa Jeong Seo, Seog Chung Seo, and Seong Oun Hwang. Efficient implementation of AES-CTR and AES-ECB on GPUs with applications for high-speed FrodoKEM and exhaustive key search. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(6):2962–2966, 2022.

[Mar07] Gary Marks. Introduction: Triangulation and the square-root law. *Electoral Studies*, 26(1):1–10, 2007.

[Mes00] Thomas S. Messerges. Using second-order power analysis to attack DPA resistant software. In *Cryptographic Hardware and Embedded Systems — CHES 2000*, pages 238–251. Springer Berlin Heidelberg, 2000.

[Mey12] Olivier Meynard. *Characterization and use of the EM radiation to enhance side channel attacks.* PhD thesis, Télécom ParisTech, 2012.

[MOW14] Luke Mather, Elisabeth Oswald, and Carolyn Whitnall. Multi-target DPA attacks: Pushing DPA beyond the limits of a desktop computer. In *Advances in Cryptology – ASIACRYPT 2014*, pages 243–261. Springer Berlin Heidelberg, 2014.

[MPO05] Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully attacking masked AES hardware implementations. In *Cryptographic Hardware and Embedded Systems – CHES 2005*, pages 157–171. Springer Berlin Heidelberg, 2005.

[MPP16] Houssem Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. Breaking cryptographic implementations using deep learning techniques. In *Security, Privacy, and Applied Cryptography Engineering*. Springer Int. Publishing, 2016.

[MS16] Amir Moradi and François-Xavier Standaert. Moments-correlating DPA. In *Proceedings of the 2016 ACM Workshop on Theory of Implementation Security*, page 5–15. Association for Computing Machinery, 2016.

[nrf] nRF52832 product specification. `https://docs.nordicsemi.com/bundle/ps_nrf52832/page/nrf52832_ps.html`.

[OC16] Colin O'Flynn and Zhizhang Chen. Power analysis attacks against IEEE 802.15.4 nodes. In *Constructive Side-Channel Analysis and Secure Design*, pages 55–70. Springer International Publishing, 2016.

[OGOP04] S.B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel. Power-analysis attack on an ASIC AES implementation. In *International Conference on Information Technology: Coding and Computing 2004*, volume 2, pages 546–552, 2004.

[O4] Colin O'Flynn. Phase modulation side channels: Jittery JTAG for on-chip voltage measurements. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2024(4):382–424, 2024.

[PH18] Christophe Pfeifer and Patrick Haddad. Spread: a new layer for profiled deep-learning side-channel attacks. Cryptology ePrint Archive, Paper 2018/880, 2018.

[PSB+18] Emmanuel Prouff, Remi Strullu, Ryad Benadjila, Eleonora Cagli, and Cécile Canovas. Study of deep learning techniques for side-channel analysis and introduction to ASCAD database. Cryptology ePrint Archive, Paper 2018/053, 2018.

[RD20] Mark Randolph and William Diehl. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptography*, 4(2), 2020.

[RSWO17] Eyal Ronen, Adi Shamir, Alisa Weingarten, and Colin O'Flynn. IoT goes nuclear: Creating a ZigBee chain reaction. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 195–212. IEEE, 2017.

[SS10] Peter J Schreier and Louis L Scharf. *Statistical signal processing of complex-valued data: the theory of improper and noncircular signals*. Cambridge university press, 2010.

[tin] Tiny AES in C. `https://github.com/kokke/tiny-AES-c/`.

[Tiu05] Chin Chi Tiu. *A new frequency-based side channel attack for embedded systems*. PhD thesis, University of Waterloo, 2005.

[usr] USRP-2944 specifications. `https://www.ni.com/docs/en-US/bundle/usrp-2944-specs/page/specs.html`.

[VMC19] Aurélien Vasselle, Philippe Maurine, and Maxime Cozzi. Breaking mobile firmware encryption through near-field side-channel analysis. In *Proceedings of the 3rd ACM Workshop on Attacks and Solutions in Hardware Security Workshop*, pages 23–32, 2019.

[WD21] Huanyu Wang and Elena Dubrova. Tandem deep learning side-channel attack on FPGA implementation of AES. *SN Computer Science*, 2(5):373, 2021.

[Wel62] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

[WHF03] Doug Whiting, Russ Housley, and Niels Ferguson. Counter with CBC-MAC (CCM). Technical report, 2003.

[WWD20] Ruize Wang, Huanyu Wang, and Elena Dubrova. Far field EM side-channel attack on AES using deep learning. In *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*, page 35–44. Association for Computing Machinery, 2020.

[WWDB21] Ruize Wang, Huanyu Wang, Elena Dubrova, and Martin Brisfors. Advanced far field EM side-channel attack on AES. In *ACM Workshop on Cyber-Physical System Security*, page 29–39. Association for Computing Machinery, 2021.

[YZC+17] Wei Yang, Yongbin Zhou, Yuchen Cao, Hailong Zhang, Qian Zhang, and Huan Wang. Multi-channel fusion attacks. *IEEE Transactions on Information Forensics and Security*, 12(8):1757–1771, 2017.