Side-Channel Attacks: Low-Cost Correlation Power Analysis of AES on Embedded Devices

Jörn-Marc Schmidt
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
email: joern-marc.schmidt@iu.org

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
email: alexander.lawall@iu.org

***Abstract*—Side-channel attacks exploit unintended physical leakages such as power consumption to compromise cryptographic implementations. Despite extensive research, practical experimentation typically requires costly and specialized equipment, limiting accessibility. This paper investigates whether effective power-based side-channel analysis can be achieved using low-cost hardware. The research goal is to evaluate the feasibility, effectiveness, and limitations of budget-constrained attacks. Accordingly, the study addresses the following questions: (i) Can inexpensive hardware enable reliable key recovery? (ii) What are the limitations regarding noise, sampling rate, and signal quality? (iii) How does clock frequency influence attack success? An experimental methodology based on Correlation Power Analysis (CPA) is applied to answer these questions. A Raspberry Pi Pico 2 is used to capture power traces from an ATtiny85 microcontroller executing Advanced Encryption Standard (AES) operations. The traces are analyzed using a Hamming-weight leakage model and Pearson correlation under varying clock frequencies. The results demonstrate that successful key recovery is achievable with minimal hardware, confirming the practical feasibility of low-cost side-channel attacks. However, limitations related to noise, trace alignment, and hardware constraints affect robustness. This work highlights critical security implications for embedded systems and contributes a reproducible, low-cost experimental framework.**

***Keywords*—Side-channel attacks, Correlation Power Analysis (CPA), AES, embedded systems security, hardware security.**

I. Introduction

Physical information leakage, like modern side-channel attacks on cryptographic implementations were publicly introduced by Kocher in 1996 [1]. He showed the time required for an operation involving the private key reveals information about that key for various asymmetric schemes, including the Rivest–Shamir–Adleman (RSA) algorithm, the Diffie-Hellman key agreement, and the Digital Signature Algorithm (DSA). In particular, measuring the execution time across several runs with the same key can reveal that key. In 1999, Kocher, Jaffe, and Jun showed that the power consumption of devices is influenced by the processed data, and that this information can be exploited as well to reveal cryptographic key material [2]. Since those initial publications, other side-channels, such as electromagnetic emanation [3] and temperature [4], have been studied. Researchers developed various techniques to analyze algorithms and countermeasures.

An important factor when considering side-channel analysis is the quality of the measured information. In the case of power consumption, a common way is to use a digital oscilloscope.

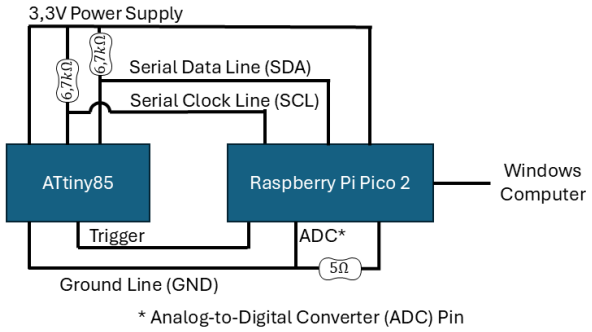

Fig. 1. Low-cost Measurement Setup for Power Side-channel Analysis with the Raspberry Pi Pico 2 and ATtiny85

One possibility is that a small resistor is added to the power supply line or ground line of the analyzed device, and the voltage drop across this resistor is measured. The measured value is proportional to the actual power consumption. However, more complex circuits have been presented as well [5]. The cost of a suitable digital oscilloscope that can store the recorded traces starts at around 70 €; advanced models can cost up to several thousand euros. An alternative is to use a specialized board for chip analysis that starts at around 50 € [6].

This work demonstrates that a power analysis is possible with an even lower cost. In particular, we used a Raspberry Pi Pico 2 [7], which is available for around 5 €, to analyze an Advanced Encryption Standard (AES) implementation running on an ATtiny85 microprocessor [8]. This demonstrates that (a) side-channel analysis is possible at low-cost without specialized equipment or a sophisticated setup and (b) our setup, which provided a resolution of 2 MS/s (Mega-Samples per Second), allows analysis of a chip running at 1, 2, or 4 MHz.

The remainder of the paper is organized as follows. Section II gives a brief overview of the foundations of the analysis. The details of the test setup are given in Section III, before the results are presented in Section IV. Section V discusses the implications of the results and limitations of the experiments. Conclusions are drawn in Section VI.

Contribution and Positioning: While Correlation Power Analysis (CPA) is a well-established technique, existing experimental work typically relies on either professional oscil-

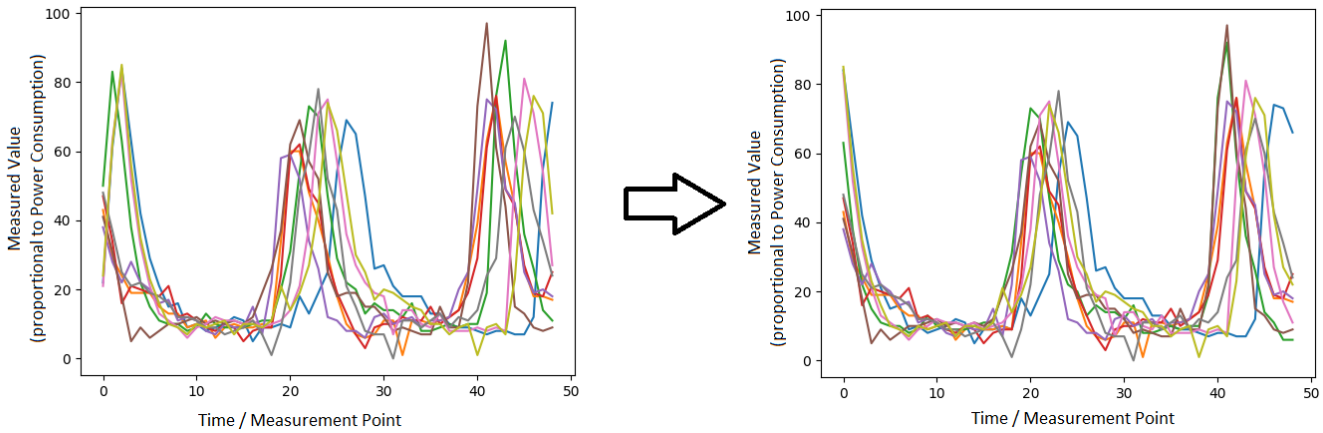

Fig. 2. Power Traces before and after Alignment (Traces are aligned by shifting the first peak to reduce temporal misalignment)

loscopes or dedicated side-channel platforms, such as ChipWhisperer. In contrast, this work investigates the lower bound of practical attack feasibility by using commodity hardware in the sub-10 € range without any dedicated measurement peripherals. Compared to prior low-cost approaches, our contribution is threefold: (i) we demonstrate that reliable key recovery is achievable using a general-purpose microcontroller with a constrained analog-to-digital converter, (ii) we systematically analyze the impact of clock frequency and sampling limitations on attack success, and (iii) we provide a reproducible experimental framework that highlights the trade-offs between cost, signal quality, and attack reliability. This positions our work at the intersection of applied cryptanalysis and embedded systems security, emphasizing that the barrier to entry for practical side-channel attacks is significantly lower than commonly assumed.

II. Background and Theoretical Foundations

The publication of the power analysis attack by Kocher, Jaffe, and Jun [2] revealed that the power consumption of a device is related to the data it processes. Brier, Clavier, and Olivier formalized the model and introduced the idea of using the Pearson correlation together with a leakage model [9]. The idea is to measure and record the power consumption of a device while it performs operations with the secret key for various different known inputs (or outputs). Each measurement is called a trace. Next, parts of the key are guessed. This hypothesis and the input give, together with the leakage model, an estimated power consumption for every recorded trace. Note that not the absolute power consumption but the relation between the different traces is important. If, for a specific guess, the correlation between the estimated power consumption and the recorded values is significantly larger than that of all other hypotheses, the guess is correct.

In our case, we analyzed an implementation of the AES algorithm [10]. It is a block cipher with a Substitution-Permutation Network (SPN) structure that encrypts 16-byte blocks. For a 128-bit key, 10 rounds consisting of the functions AddRoundKey, SubBytes, MixColumns, and ShiftRows are carried out. The last round skips the MixColumns and applies a final AddRoundKey, outputting the ciphertext. For our analysis, only the first AddRoundKey and SubBytes operations are relevant. The input bytes are XORed with the first round key, which is the input key of AES, i.e., no transformation or key schedule is applied to it. The SubBytes operation takes every byte of the result and maps it to a corresponding value from a fixed nonlinear S-box. Due to the strong nonlinearity of the AES S-box, an incorrect key-byte hypothesis — even if it differs by only one bit — results in an output that is statistically unrelated to the correct one. The analysis we performed proceeded as follows. Here, the goal is to analyze the key byte at position 0, Key_0:

1) Record N traces of length l, i.e., for $n = 0, \ldots, N-1$, the trace points $t_0^n \ldots, t_l^n$ and the corresponding input texts $input_0^n, \ldots, input_{15}^n$
2) Consider all 256 possibilities for key byte Key_0, i.e., $guess_0 = 0..255$. For every guess,
   a) compute for $n = 0, \ldots, N-1$ the intermediate value $v^n = \text{S-box}(guess_0 \oplus input_0^n)$ and
   b) apply the leakage model, e.g., for the Hamming-weight compute $hw^n = HammingWeight(v^n)$.
   c) For every position $p = 1, \ldots, l$ of the trace, compute the correlation between $(t_p^0, \ldots, t_p^{N-1})$ and $(hw^0, \ldots, hw^{N-1})$ and take the maximum over all positions.
3) Assuming sound recordings, a well-chosen leakage model, a sufficient number of recorded traces, and detectable information leakage in the power consumption, the guess with the highest correlation value is very likely the correct one, i.e., Key_0.

After determining the first byte, the procedure can be repeated from step 2 for the next byte. That is, no new recording of traces is required.

TABLE I
EXPERIMENTAL SETUP AND MEASUREMENT PARAMETERS

Measurement device	Raspberry Pi Pico 2
ADC resolution	12 bit
Effective sampling rate	2 MS
Measurement method	Low-side shunt measurement
Shunt resistor	5 Ω
Device under test (DUT)	ATtiny85
DUT supply voltage	3.3 V
DUT clock source	Internal oscillator with prescaler
Tested frequencies	1, 2, and 4 MHz
DUT–measurement connection	I2C + trigger pin
Tested instruction	AES S-box table lookup
Triggering	GPIO trigger around a sequence of four lookups

III. EXPERIMENTAL SETUP AND METHODOLOGY

The core of our measurement setup was a Raspberry Pi Pico 2. It features a dual Arm Cortex-M33 processor, a 520 kB on-chip volatile memory and a Universal Serial Bus (USB) 1.1 controller. According to the datasheet, the Pico 2 comes with 4 General-Purpose Input/Output (GPIO) pins that can be used for analog-to-digital conversion and a 12-bit 500 kS/s Analog-to-Digital Converter (ADC) [11]. However, increasing the system clock frequency and configuring the ADC to use this source, as described by Roberts [12], allowed recording at 2 MS/s.

The device under test was an ATtiny85, an 8-bit microcontroller [8]. It comes with an internal clock source that allows running the device at 1, 2, as well as 4 MHz; other options up to 16 MHz are also possible, but they were not considered in our analysis.

Both chips were placed on a breadboard and connected as shown in Figure 1. The ATtiny was supplied with 3.3 V from the Raspberry Pi. Communication between the Raspberry Pi and the ATtiny was carried out via I2C. Two resistors of 6.7 kΩ were used as pull-ups for the I2C connection, i.e., for the Serial Data Line (SDA) and the Serial Clock Line (SCL). For the measurement, the ground line of the ATtiny was connected to the Raspberry Pi ground through a 5 Ω resistor. The ADC pin of the Raspberry Pi was connected to the ground of the ATtiny. Hence, this setup measures the voltage drop across the resistor, which is proportional to the current. The drop is related to the power consumed by the device, which is sufficient for our analysis. A trigger pin from the ATtiny to the Raspberry Pi indicated when to start and stop a measurement. The whole setup was controlled by a Windows computer that was connected to the Raspberry Pi via USB. A summary of the devices and used parameters is given in Table I.

On the ATtiny we implemented AddRoundKey and SubBytes operations with a key and input text provided by the Raspberry Pi. The S-box was stored in the flash memory of the device as a lookup table. In particular, 16 bytes of text and 16 bytes of key material were supplied by the Raspberry Pi. In case no further input was given, the current key and the output of the previous operation were used as inputs for subsequent operations. The ATtiny waited for a start signal from the Raspberry Pi, performed the computation, and returned the results upon completion. Our analysis targeted four consecutive lookup operations. The trigger pin was set after XORing the key with the input. In order to prevent interference between the trigger signal and the actual operations, after setting the trigger pin and before unsetting it, 20 and 60 No Operation (NOP) operations were performed, respectively. In between, the four S-box table lookups were performed. The Raspberry Pi controlled the setup, i.e., it set the input and key material and verified the output. After initiating the operations on the ATtiny, the Raspberry Pi waited until the trigger went high and recorded samples from the ADC pin until it went low again. The input used, together with the measured points on the ADC, was sent via USB to the computer, where a Python script collected the data and stored it for further analysis.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate our setup, we analyzed the same ATtiny85 implementation of the S-box lookup on a chip running at 1, 2, and 4 MHz using the internal oscillator. Although the oscillator runs at 8 MHz, we set a prescaler value to achieve the desired target frequency.

Visual inspection of the recorded traces revealed that, despite the trigger, they were not well aligned. Hence, in the first step, all traces were aligned. In particular, we selected the first few points of every trace and considered only traces with a peak above a threshold within this window. For traces where more than one point exceeds the threshold, the first point was used for alignment. Both the threshold and the number of points that were considered were determined by visual inspection of the first 10 traces. In particular, for 1 MHz, we selected a window between points 0 and 40 with a threshold of 65. For 2 MHz and 4 MHz, we used points 0 to 10 with a threshold of 60 and 0 to 6 with a threshold of 45, respectively. This peak was shifted to be the first point of the trace; traces without such a peak were discarded. Figure 2 shows 10 sample traces recorded at a 4 MHz clock frequency before and after the alignment process. The figure already shows that this very simple alignment method does not completely address the issue of clock jitter. However, it improved the overall result. To quantify this, we calculated the ratio between the maximum correlation of the correct hypothesis and the maximum correlation among all other key hypotheses, hereafter referred to as the Signal-to-Ghost ratio. Without this alignment, the Signal-to-Ghost ratio for Byte 0 at 4 MHz using a window size of 3, which showed the best result in our tests, dropped from 2.38 to 1.13.

To ensure comparable results, traces that had a smaller number of measurement points, e.g., due to misalignment or buffering issues with the Raspberry Pi, as well as traces that did not contain a peak in the expected window, were discarded. After removing those outliers, we analyzed 100,000 traces from each measurement that were successfully aligned.

The analysis used a Hamming-weight leakage model, as it provided better results than the Hamming distance model. In addition to the basic alignment, we used windowing of

TABLE II
CORRELATION RESULTS FOR THE FIRST TWO AES S-BOX OPERATIONS UNDER VARYING FREQUENCIES AND WINDOW SIZES

1st Byte															
	Signal-to-Ghost ratio / Correct hypothesis / Best wrong hypothesis														
Window	0			2			3			4			5		
1 MHz	0.76	0.0093	0.0132	1.10	0.0143	0.0130	0.99	0.0142	0.0144	1.02	0.0140	0.0137	1.00	0.0139	0.0139
2 MHz	1.26	0.0178	0.0141	1.27	0.0183	0.0144	1.39	0.0196	0.0141	1.46	0.0200	0.0137	1.47	0.0203	0.0138
4 MHz	2.37	0.0466	0.0197	2.27	0.0459	0.0202	2.38	0.0475	0.0200	2.32	0.0497	0.0214	2.32	0.0659	0.0284

2nd Byte															
	Signal-to-Ghost ratio / Correct hypothesis / Best wrong hypothesis														
Window	0			2			3			4			5		
2 MHz	1.00	0.0169	0.0169	1.09	0.0180	0.0165	1.10	0.0182	0.0166	1.03	0.0184	0.0178	0.93	0.0182	0.0195
4 MHz	1.21	0.0311	0.0256	1.33	0.0321	0.0242	1.34	0.0390	0.0291	1.33	0.0424	0.0320	1.41	0.0514	0.0365

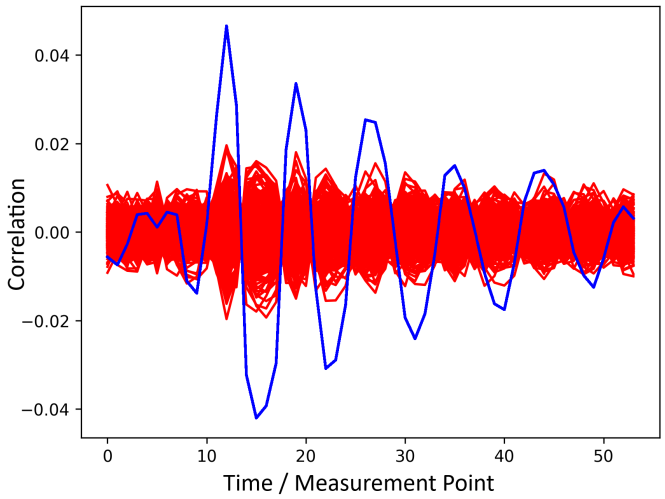

Fig. 3. Pearson correlation of the Hamming-weight of the S-box output for all possible hypotheses for the 1st key byte and the measured values. The correct key hypothesis is shown in blue.

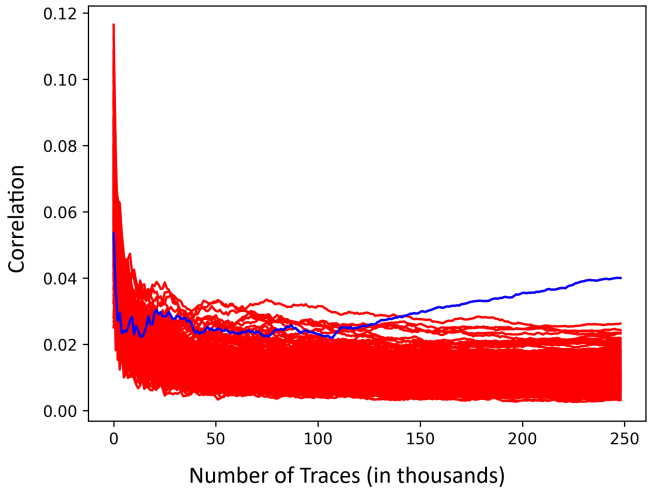

Fig. 4. Maximum correlation as a function of the number of traces for all possible hypotheses for the 3rd key byte. The correct key is shown in blue.

different sizes. That is, the points in this window were added up and divided by the window size. In this way, the noise was reduced to achieve a better Signal-to-Noise Ratio (SNR). In addition, this approach helped mitigate clock jitter, as the internal clock was not as precise as an external clock generator would have been.

For all three clock frequencies, we were able to determine the first key byte correctly. For 2 and 4 MHz, the second key byte could also be determined. Figure 3 shows the correlation for all key hypotheses between the measurements at 4 MHz and the first S-box output. The correct hypothesis is shown in blue. The measurements were aligned to the first peak, but no windowing was applied. The results for all three different clock frequencies for the first two bytes are given in Table II, besides the 2nd byte for 1 MHz, at it could not determined successfully. The table shows that the best results were achieved at 4 MHz and that not all setup configurations and window sizes were able to identify the correct key byte reliably. If the Signal-to-Ghost ratio was 1 or below, the correct hypothesis did not stand out.

For the 3rd key byte, different alignment parameters were tested. For a frequency of 1 MHz, the 3rd byte could not be determined successfully. For 2 and 4 MHz, the results indicate that aligning the traces using the 2nd peak as reference yielded the best results. For 2 MHz, selecting a peak between points 6 and 14 with a threshold of 60 resulted in a Signal-to-Ghost ratio of 1.35 without windowing and 1.39 for a window size of 2 points. For 4 MHz, aligning the traces to the peak between points 7 and 14 without windowing yielded the highest correlation. Figure 4 shows the evolution of the maximum correlation as a function of the number of measurements for the 3rd key byte. For each trace count, the correlation was computed for all 256 key hypotheses and the maximum value over time samples was retained. The correct key hypothesis is highlighted in blue. It illustrates that for around 150,000 traces, the correct key hypothesis became statistically distinguishable.

V. DISCUSSION, IMPLICATIONS, AND LIMITATIONS

Side-channel leakage and how to exploit it to reveal secrets of cryptographic devices have been extensively studied in

academia and industry for many years. Various techniques, including CPA, have been documented in standard references such as the book by Mangard et al. [5]. However, most documented CPA-related experiments rely on professional measurement equipment or dedicated side-channel platforms, which imposes a non-negligible cost and expertise barrier. In contrast, our results show that it is possible to measure power leakage at very low cost without such specialized equipment. Hence, our contribution is not a new attack technique, but an investigation on practical lower bounds for power-based side-channel analysis.

The threat model considered in this work assumes an attacker has temporary physical access to the target device but is not equipped with sophisticated equipment. Such an attacker might be involved in device resale/repair or gains temporary access to smart home or other IoT devices. Using our low-cost setup for the attack comes at the cost of a low sampling rate. In the measurements, subsequent operations interfere with each other. The data used in operations that were carried out directly after the sequence of NOP operations could be clearly identified. Although the measurements at $4\,\mathrm{MHz}$ recorded only one point per two clock cycles, they showed better results than those at 1 and $2\,\mathrm{MHz}$. However, the tests at $8\,\mathrm{MHz}$ did not yield statistically significant results. Nevertheless, our results demonstrate that effective side-channel analysis remains feasible at low-cost under certain conditions. Our evolution across different clock frequencies highlights how undersampling, noise, and clock instability influence attack reliability. In particular, when considering higher clock frequencies, our simple setup might no longer be sufficient.

When analyzing real-world devices, the limited clock frequency and the interference between operations are not the only obstacles. Our setup used a trigger to identify the region of interest; in other applications, this may not be possible, and locating the relevant interval may require scanning for specific patterns in the power consumption. In addition, our measurements were conducted directly at the chip. When analyzing chips that are part of an application, the power supply might not be directly accessible, and/or the boards used may include capacitors to stabilize the power supply that either affect the measurements or need to be removed.

Future work will focus on ways to improve the setup, for example by soldering the components together instead of using a simple breadboard and by analyzing the extent to which an operational amplifier can improve the measurements. When using an LM358-N, the chip introduced so much noise that the SNR decreased. We therefore plan to investigate alternatives that improve the setup while remaining cost-effective.

VI. Conclusion and Future Work

This paper demonstrated that Correlation Power Analysis can be effectively conducted using low-cost and widely accessible hardware. The experimental results confirm that even constrained measurement environments provide sufficient information leakage to enable practical key recovery from embedded AES implementations. These findings reinforce the notion that side-channel vulnerabilities are not limited to high-end laboratory settings but remain a realistic threat in resource-constrained attack scenarios. Beyond feasibility, the study highlights the critical role of signal quality and preprocessing in low-cost environments. The effectiveness of trace alignment and noise mitigation techniques underscores that methodological rigor can partially compensate for hardware limitations. At the same time, the observed constraints emphasize that attack reliability remains sensitive to measurement conditions and device characteristics. From a security standpoint, the results stress the necessity of integrating side-channel resistance into embedded system design, even for low-cost devices.

Future work will focus on enhancing measurement robustness while preserving cost efficiency. Promising directions include improved hardware integration, advanced signal processing techniques, and the extension of the approach to more complex and less controlled real-world scenarios, particularly in the absence of explicit trigger signals.

References

[1] P. C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, ser. CRYPTO '96. Berlin, Heidelberg: Springer-Verlag, 1996, p. 104–113.

[2] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, M. Wiener, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[3] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2001, pp. 251–261.

[4] M. Hutter and J.-M. Schmidt, "The temperature side-channel and heating fault attacks," in *Smart Card Research and Advanced Applications: 12th International Conference, CARDIS 2013, Berlin, Germany, November 27-29, 2013. Revised Selected Papers 12*. Springer, 2013, pp. 219–235.

[5] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, ser. Advances in Information Security. New York, NY: Springer, 2007, vol. 31.

[6] NewAE Technology, "CW1101 ChipWhisperer-Nano," https://chipwhisperer.readthedocs.io/en/latest/Capture/ChipWhisperer-Nano.html, retrieved: 11th March, 2026.

[7] Raspberry Pi, "Raspberry Pi Pico 2," https://www.raspberrypi.com/products/raspberry-pi-pico-2/, retrieved: 11th March, 2026.

[8] Microchip, "ATtiny85," https://www.microchip.com/en-us/product/attiny85, retrieved: 11th March, 2026.

[9] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Springer, 2004, pp. 16–29.

[10] National Institute of Standards and Technology (NIST), "Advanced encryption standard (AES)," National Institute of Standards and Technology, Federal Information Processing Standards Publication FIPS PUB 197-upd1, May 2023. [Online]. Available: https://doi.org

[11] Raspberry Pi Ltd, "Raspberry Pi Pico 2 Datasheet," https://pip-assets.raspberrypi.com/categories/1005-raspberry-pi-pico-2/documents/RP-008299-DS-1-pico-2-datasheet.pdf, 2024.

[12] S. Roberts, "Quadrupling the sample rate of an RP2040 oscilloscope," https://www.hackster.io/sandy-roberts2/quadrupling-the-sample-rate-of-an-rp2040-oscilloscope-5f9a74, retrieved: 11th March, 2026.