Generation of Synthetic Power Traces for Power Analysis Attacks

Student Name: Dhruv Visariya
Supervisor: Prof. Rajneesh Kumar Srivastava
Department: Electronics and Communication Engineering
Institute: University of Allahabad
Semester: 8th Semester

Abstract

Side-channel analysis, particularly power analysis, remains a formidable threat to cryptographic implementations. Traditional trace acquisition methods often require expensive hardware setups and noise-isolated environments. This project addresses this challenge by introducing a synthetic trace generation approach guided by Points of Interest (POI) analysis. Leveraging real power traces from the DPAcontest v2 dataset, key operations in AES encryption are temporally segmented and modeled using HD (Hamming Distance) and HW (Hamming Weight) leakage models. Through statistical validation using Pearson correlation and Cohen's d effect size, we show that synthetic traces can replicate key characteristics of real-world leakage patterns. The proposed method offers a practical, lightweight framework for researchers to study power leakage behavior without relying on costly hardware setups.

---

1. Introduction

Power analysis attacks exploit the physical emissions of cryptographic hardware to retrieve secret keys. While mathematically secure, cryptographic algorithms like AES can leak information through variations in power consumption. Extracting meaningful information from such leakage typically requires high-resolution oscilloscopes, synchronized triggering, and low-noise environments. For many academic and student researchers, this requirement poses a barrier.

This work explores an alternative: synthetic power trace generation. Rather than depending on hardware captures, it relies on analyzing real traces to extract Points of Interest (POIs), map them to Hamming models, and reconstruct synthetic segments through interpolation. This enables low-cost experimentation with trace analysis and countermeasure development.

2. Background & Related Work

Side-channel attacks can be classified as passive, non-invasive methods that leverage leakages such as power, timing, or electromagnetic emissions. Kocher et al. [1] formalized Differential Power Analysis (DPA), revealing that data-dependent leakage can be statistically exploited to extract keys.

CMOS circuits leak power primarily during switching events. Models like HW (number of bits set) and HD (bit transitions between values) are commonly used to quantify data-dependence [1,8]. Points of Interest (POIs), typically extracted using statistical tools or derivatives, are time samples where the leakage is most prominent.

While SPICE simulations and RTL-based tools exist for modeling leakage, they often fail to capture realistic noise or operation jitter. Public datasets like DPAcontest v2 [7] offer high-quality real traces, but mask variations and preprocessing steps hinder their use in learning-based leakage mapping. Our work bridges this gap by aligning real trace characteristics with interpretable POI structures.

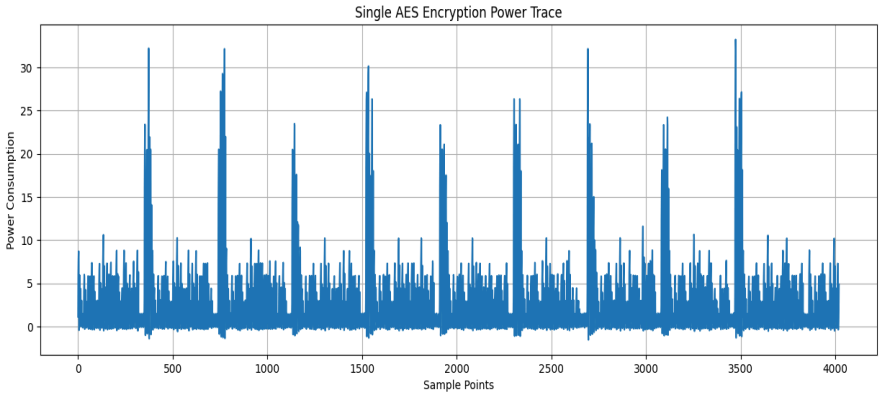
---

3. Methodology

3.1 Simulation Attempt & Limitations

Initially, AES operations were simulated using a lumped-capacitance model. Each operation (e.g., SubBytes, MixColumns) was assigned a fixed time segment, and modeled using exponential discharge curves and sinusoidal clock overlays. However, all operations produced nearly identical waveforms, regardless of input data. This lacked HW/HD sensitivity and realistic jitter.

FIGURE : Output of Initial Simulation Approach-



3.2 Real Trace Analysis from DPAcontest v2

The DPAcontest v2 dataset [7] contains unmasked AES-128 traces suitable for empirical leakage study. Using visual inspection and SPA-based segmentation, we identified repeatable segments corresponding to key operations. For instance, SubBytes consistently occurred around 77.3 ns to 84.1 ns across traces.

FIGURE : AES round segregation based on repetitive pattern-

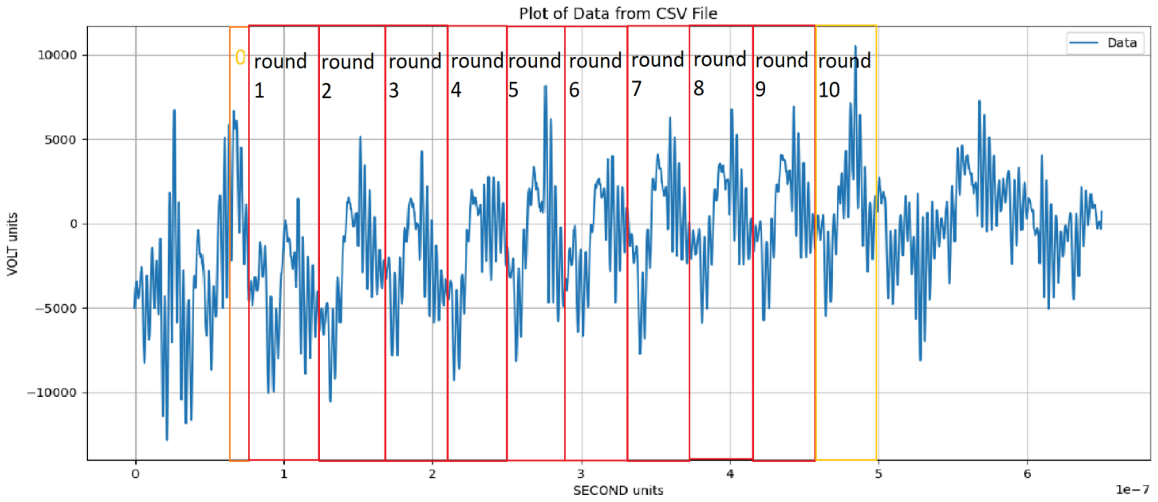
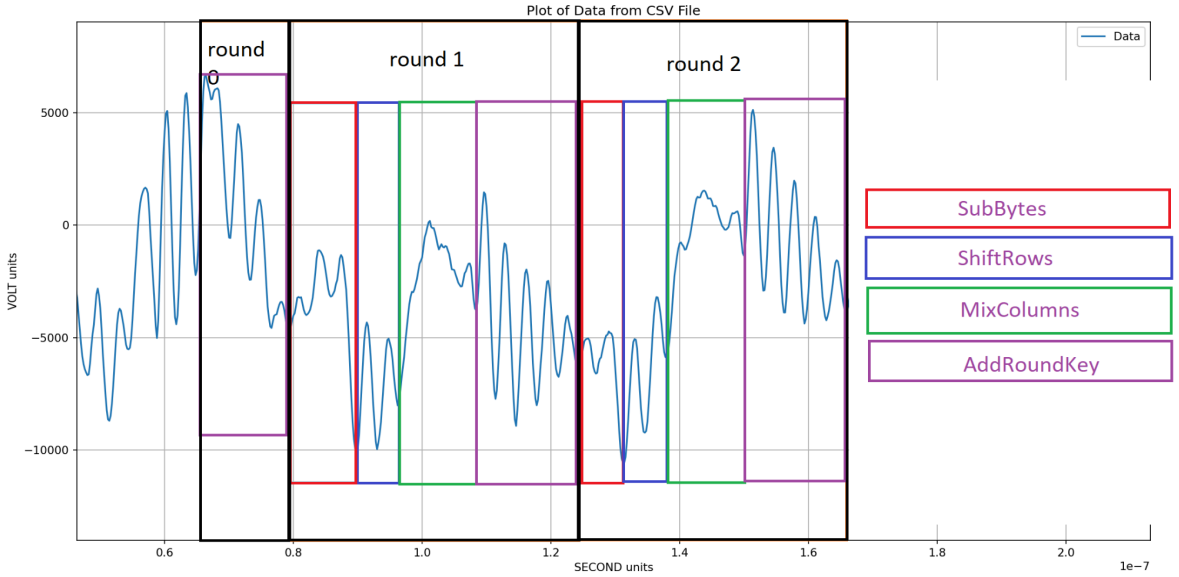


FIGURE: Segregation of power signature of operations in AES round



3.3 Points of Interest (POI) Extraction and Interpolation

We used the first and second derivatives of power traces to detect POIs. Inflection points, zero-crossings, and sharp changes were retained. Each segment was interpolated using:

- **Cubic Spline Interpolation** for smoothness.
- **Chebyshev Polynomial Approximation** to reduce endpoint oscillations.

These interpolated segments approximated the real signal well and were used to generate synthetic traces.

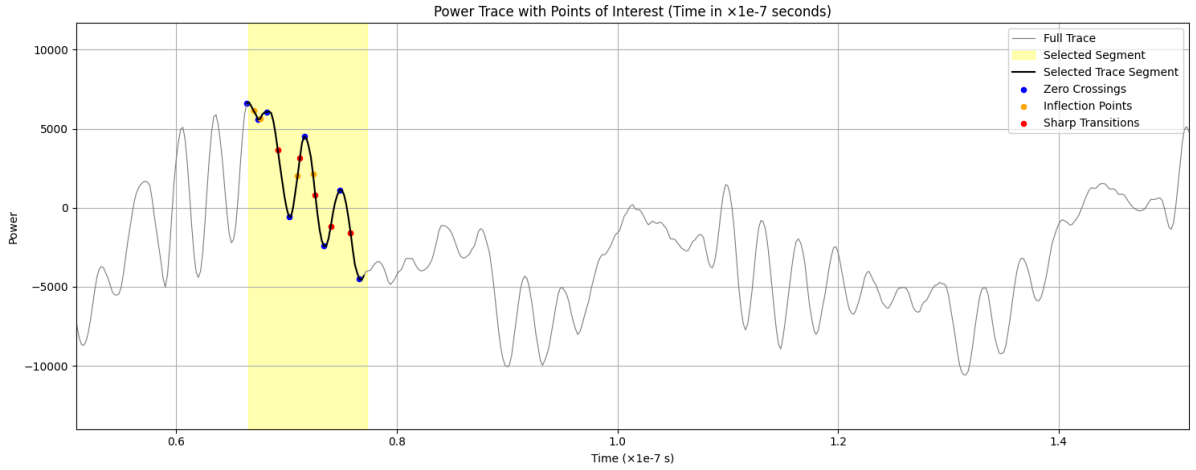


FIGURE:POI DETECTION IN DEDICATED REGION

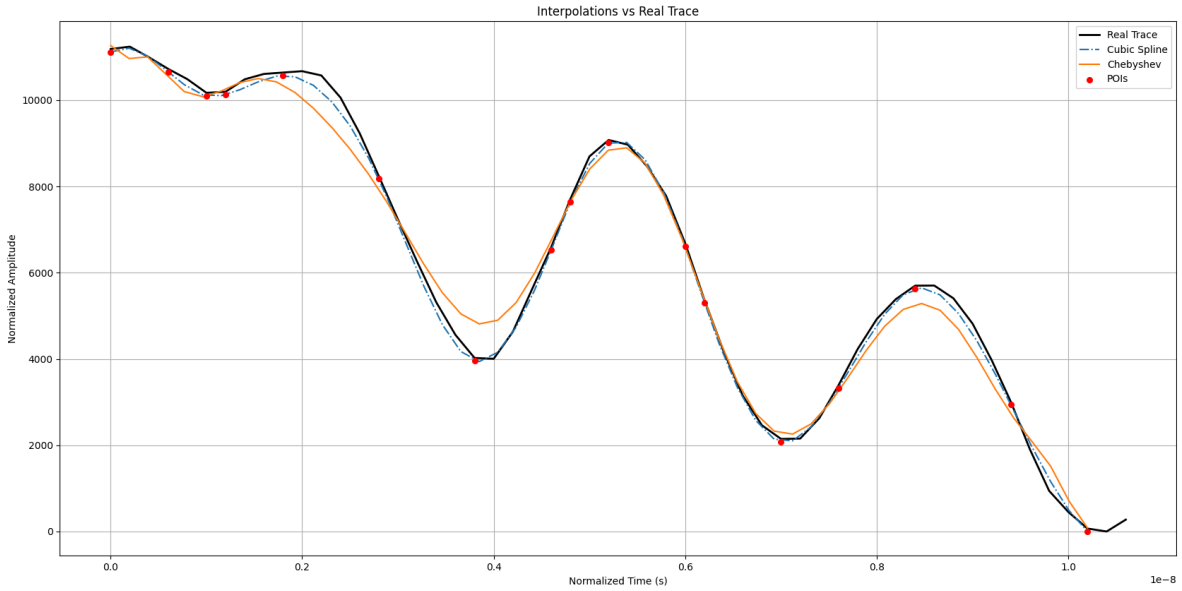


Figure : a comparison plot of (a) a real trace segment, (b) POI locations marked on it, and (c) its interpolated version using Cubic Spline and Chebyshev methods.

3.4 Mapping to HD/HW Models

Each trace's POI amplitude was mapped to the corresponding HD and HW values derived from XOR operations between key and plaintext. This enabled construction of template segments parameterized by data-dependent features.

---

4. Evaluation and Results

4.1 Intra-Group Similarity: Pearson Correlation Coefficient

To assess the consistency of power traces within the same class of data transitions, the **Pearson correlation coefficient (r)** was computed between all possible trace pairs within each Hamming Distance (HD) and Hamming Weight (HW) group. This measure evaluates how similarly two time-aligned traces vary, capturing the linear similarity of power consumption patterns.

The results revealed strong intra-group correlations, particularly in HD groups where **mean r values consistently exceeded 0.95**. This strongly validates the hypothesis that operations processing data with similar HW/HD exhibit reproducible and distinguishable power patterns—a key assumption behind Simple Power Analysis (SPA) and Differential Power Analysis (DPA) models.

> *Example*: For HD=53, the mean intra-group correlation reached **r ≈ 0.995**, confirming near-identical power characteristics across multiple traces.

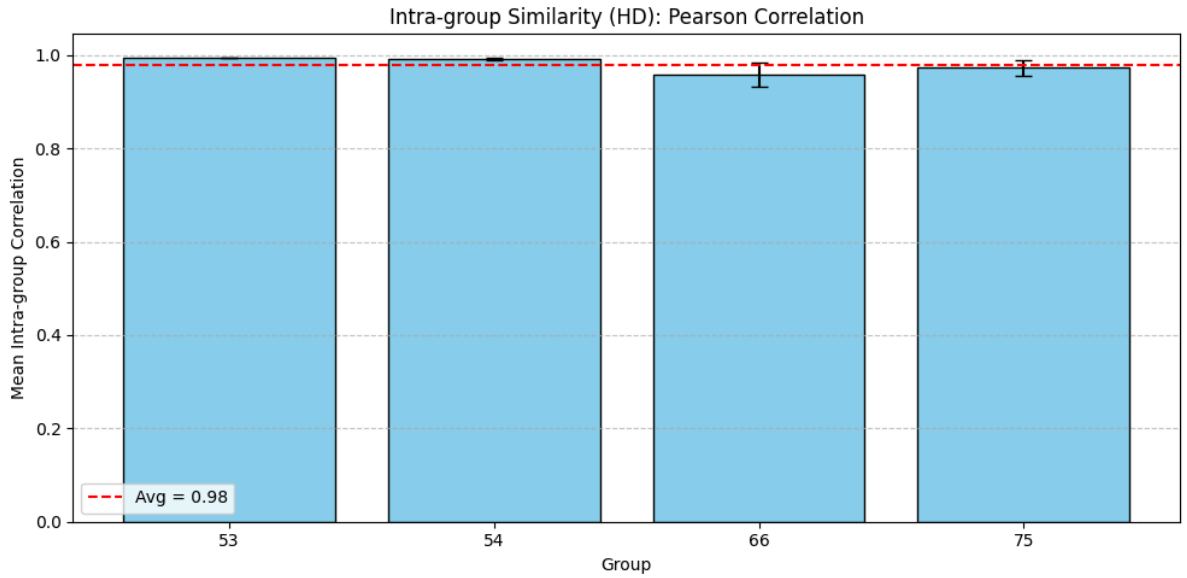


FIGURE : Boxplot or violin plot of intra-group Pearson correlation for each HD group

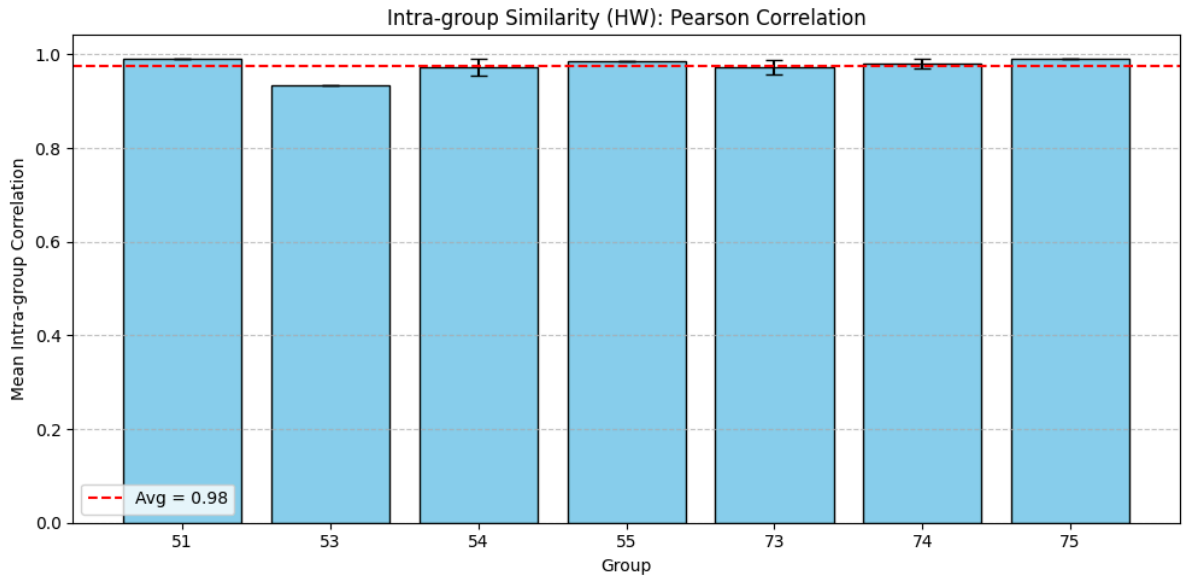


FIGURE :Boxplot or violin plot of intra-group Pearson correlation for each HW group

4.2 Inter-Group Differences: Cohen's d Effect Size

While intra-group similarity confirms trace stability, effective side-channel models also require that **different classes produce distinguishable patterns**. For this, the **Cohen's d effect size** was computed across all group pairs, measuring the standardized difference in POI amplitudes.

Cohen's d was evaluated **point-wise across all POIs**, and group pairs were considered significantly distinguishable if at least **5 consecutive POIs exhibited |d| > 0.8**, following established SCA literature heuristics.

Result Highlights:

- **HD=53 vs HD=75** showed |d| > 1.4 across 30+ POIs
- **HW group comparisons** also revealed high effect sizes in many pairs (e.g., HW=51 vs HW=75 with **|d|max > 10**), suggesting strong data-driven leakage variations.
- This effect-size-based approach goes beyond average differences by capturing localized and consistent divergence across traces, making it ideal for evaluating **template-based leakage models**.

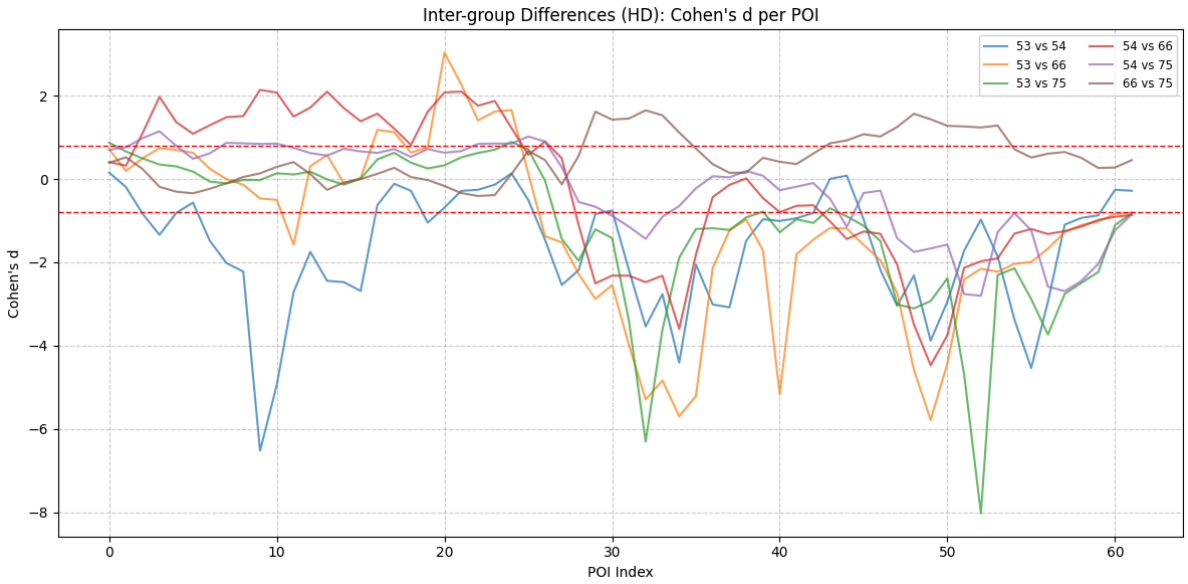


FIGURE 5.3 : Line plot of Cohen's d over POIs for selected HD group pairs

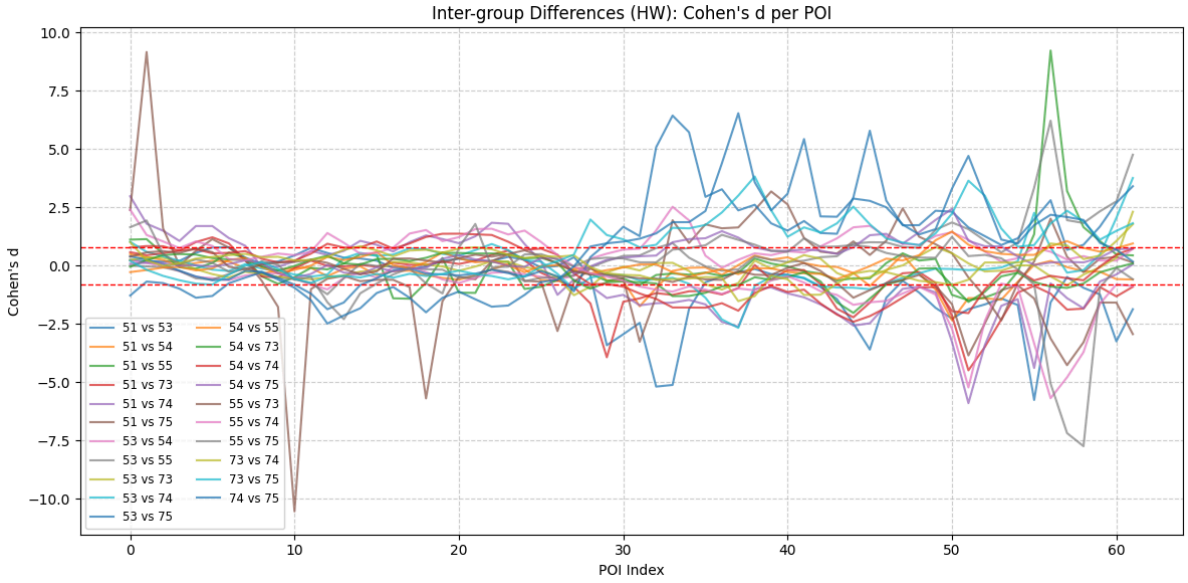


FIGURE : Line plot of Cohen's d over POIs for selected HW group pairs

These results indicate that despite some noise and overlap, many groups remain statistically distinguishable — a key condition for successful classification or key recovery.

5. Conclusion & Future Work

This work demonstrates that realistic synthetic power traces can be generated using POI-driven interpolation methods. The strong intra-group correlation and measurable inter-group differences validate the HD/HW-based modeling hypothesis. The approach reduces reliance on hardware and facilitates accessible research.

Future Directions:

- Automated POI selection using statistical thresholds or ML
- Modeling delay segments and control flow artifacts in traces
- Leveraging synthetic traces to test countermeasures (masking/hiding)
- Expanding to other ciphers and datasets

References

[1] Mangard, S., Oswald, E., & Popp, T. (2007). Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer.

[2] Harris, C. R., et al. (2020). Array programming with NumPy. Nature.

[3] Virtanen, P., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. Nature Methods.

[4] McKinney, W. (2010). Data Structures for Statistical Computing in Python. PyData.

[5] Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering.

[6] Waskom, M. (2021). Seaborn: Statistical data visualization. JOSS.

[7] DPAcontest v2 Dataset. Retrieved from https://www.dpacontest.org/v2/

[8] Rabaey, J. M., Chandrakasan, A., & Nikolic, B. (2003). Digital Integrated Circuits: A Design Perspective. Prentice Hall.