The Thesis Committee for Amit Kumar
Certifies that this is the approved version of the following thesis:

Simulation-Based Verification of EM Side-Channel Attack Resilience of Embedded Cryptographic Systems

APPROVED BY

SUPERVISING COMMITTEE:

Michael Orshansky, Supervisor

Ali Yilmaz

Simulation-Based Verification of EM Side-Channel Attack Resilience of Embedded Cryptographic Systems

by

Amit Kumar, B.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2017

Abstract

Simulation-Based Verification of EM Side-Channel Attack Resilience of Embedded Cryptographic Systems

Amit Kumar, M.S.E.
The University of Texas at Austin, 2017

Supervisor: Michael Orshansky

Electromagnetic (EM) fields emanated due to switching currents in crypto-blocks can be an effective non-invasive channel for extracting secret keys. Accurate design-time simulation tools are needed to predict vulnerabilities and improve resilience of embedded systems to EM side-channel analysis attacks. Modeling such attacks is challenging, however, as it requires a multitude of expensive simulations across multiple circuit abstraction levels together with EM simulations. In this work, a simulation flow is developed to study the differential EM analysis (DEMA) attack on the Advanced Encryption System (AES) block cipher.

The proposed flow enables design-time evaluation of realistic DEMA attacks for the first time. The major challenge is accurately computing signals

received by a nearby probe at various positions above the chip surface for a large number of AES encryptions. This requires rapidly generating spatial distribution and transient EM radiation of on-chip current waveforms. Commercial CAD tools are used to generate space-time samples of these waveforms and a custom EM simulator to radiate them. The computations are sped up by focusing on information-leaking time windows, performing hybrid gate- and transistor-level simulations, radiating only the currents on top metallization layers, and generating traces for different encryptions in parallel. These methods reduce simulation time to a manageable $\sim$ 20 hrs wall-clock time/attack allowing a previously impossible level of vulnerability analysis.

The proposed flow also allows pinpointing critical regions on the chip most susceptible to EM attacks. We demonstrate that exploiting the spatial profile of circuit elements can reveal cryptographic keys with significantly fewer number of traces than DPA , guiding designers to the most critical areas of the layout. This enables targeted deployment of counter-measures to the highest information-leaking design components.

Table of Contents

List of Figures

Chapter 1

Introduction

This thesis introduces a simulation flow that enables rapid and accurate design-time prediction of Electromagnetic side-channel analysis (EM-SCA)resilience of cryptographic modules for the first time. The simulation costs are reduced without sacrificing predictive value as follows. Step 1: Commercial CAD tools are used to run highly-optimized transistor-level simulations only during critical time windows when information leakage happens. Steps 2-3: The Electromagnetic (EM) radiation is limited to the currents distributed on the top-metallization layer power/ground interconnects and traces for different encryptions are generated in parallel. Using the proposed methodology, various differential attacks on Advanced Encryption Standard (AES) block cipher are simulated.

The rest of this thesis is arranged as follows. Chapter 2 discusses about AES block cipher and differential attacks in detail. In Chapter 3, prior work and current simulation methodologies are discussed along with their limitations. In Chapter 4, the proposed simulation flow for circuit analysis and EM radiation has been described. In Chapter 5, the system setup and Differential EM Attack (DEMA) results are discussed and it is shown how different design choices lead to a vulnerable system. Chapter 6 summarizes the work with its limitations and discusses future work to be done in this direction.

Chapter 2

AES and Differential Attacks

The simulation of EM SCA attacks on block ciphers requires T_{EM} traces, observable via a nearby EM field probe, to be computed during critical cipher execution steps. Once the traces are obtained, the vulnerability of the design to various EM SCA attacks can be investigated. To demonstrate the proposed simulation flow, this work focuses on the simulation of a differential analysis attack using the EM side channel to extract the secret key, termed the differential EM attack (DEMA) [1], [2], on an ASIC implementation of the Advanced Encryption Standard (AES) [3]. This chapter discusses AES block cipher and differential attacks on block ciphers.

2.1 AES Cipher

AES is based on a substitution-permutation network and has been widely used since 1999 for symmetric cryptography applications; it has a fixed block size of 128 bits and key sizes of 128, 192 or 256 bits [3]. The key size determines the number of transformation rounds input (known as plain text) goes through to generate encrypted data also known as cipher text. There are 10, 12 and 14 rounds required for 128, 192 or 256 bits respectively. Each transformation round uses a Round Key derived from the secret key using Key Expansion function.

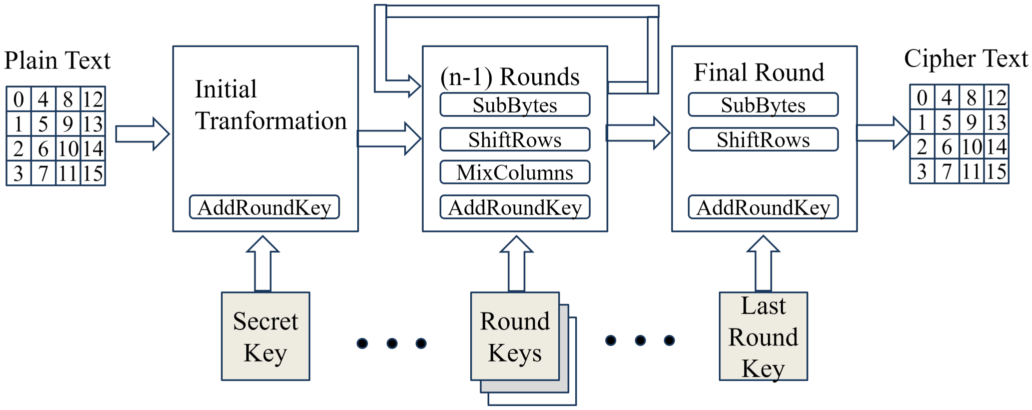
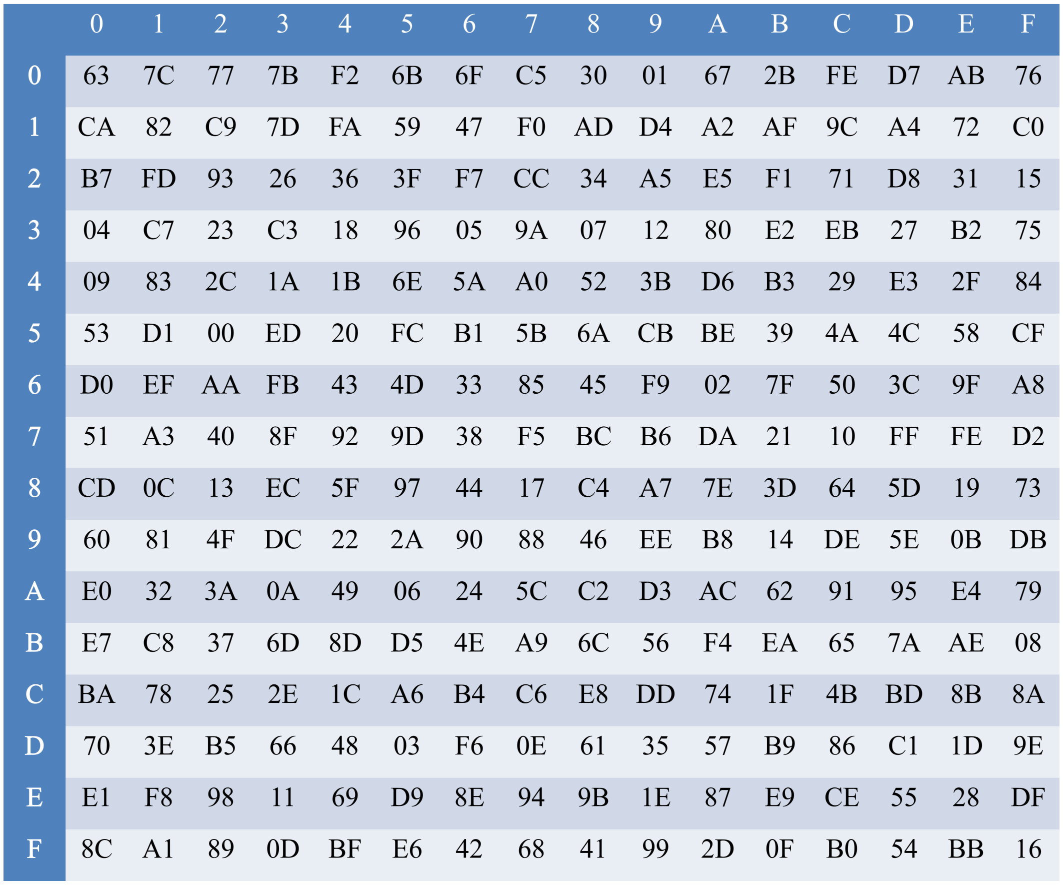


Figure 2.1: Top-level Diagram of AES

Fig. 2.1 shows the shows the order in which bytes are written into the state as plaintext and read from the state as ciphertext. The numbers in the matrices for plain text and cipher text represent the byte number with 0 being the most significant and 15 being the least significant byte. As shown in Fig. 2.5, AES is a round-based block cipher, where each round consists of four steps (the final round does not have MixColumns).

1. SubBytes: The SubBytes operation independently transforms each byte of the state. It can be implemented by a simple table lookup or can be implemented using dedicated hardware. The 16 × 16 matrix of byte values is called an S-Box as shown in Fig. 2.2. As an example, Byte 0x00 is mapped to 0x63, 0x01 to 0x7C and so on. The upper nibble is

used as the row value and lower nibble is used as column value to index into the S-Box.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	FE	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure 2.2: S-Box

2. ShiftRows: ShiftRows transformation shifts the bytes present in each row of the state matrix. Fig. 2.3 depicts this transformation. The first row is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed.

For the fourth row, a 3-byte circular left shift is performed. ShiftRows provides horizontal diffusion in the state matrix.

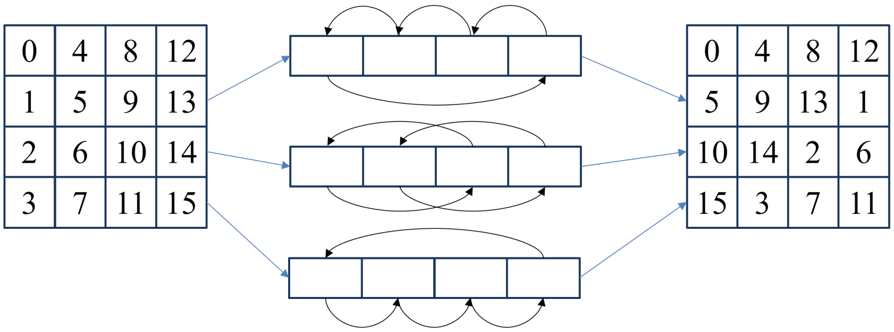


Figure 2.3: ShiftRows Transformation

3. MixColumns: MixColumns transformation maps the bytes present in each column of the state matrix to a new value. Fig. 2.4 depicts this transformation. Each byte of a column is mapped into a new value that is a function of all four bytes in that column. MixColumns provide vertical diffusion in the state matrix.

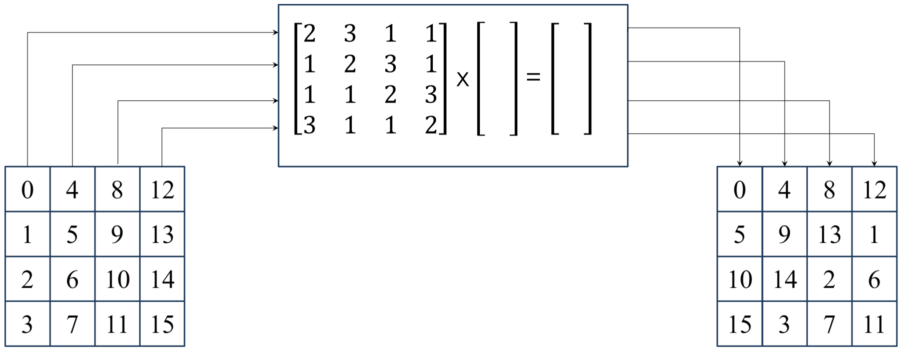


Figure 2.4: MixColumns Transformation

4. AddRoundKey: In the AddRoundKey transformation, the state matrix is bitwise XORed with the round key. The round keys are derived from the secret key using AES key expansion algorithm. For 10 rounds of transformation, the algorithms uses 128-bit secret key and 11 128-bit round keys which are used in intial round and 10 rounds of transformation.

Each round uses a Round Key derived from the secret key; hence, attackers who can get their hands on a Round Key can reverse-engineer the secret key. It is thus essential to protect the keys used in every round. In hardware implementation, the AES state is stored in a register at the end of each round. AES has been shown to leak information while computing intermediate values dependent on the secret key. One such intermediate value is present in the last round of transformation when the state register gets loaded with a new value [4].

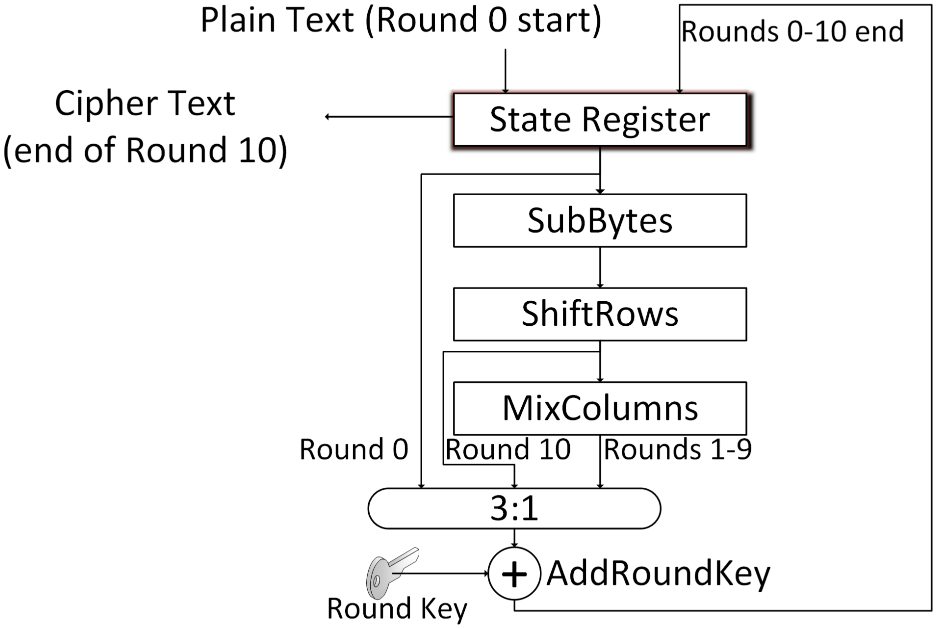


Figure 2.5: Block Diagram of 128-bit AES Cipher

2.2 Differential Side-channel Attacks

In side-channel attacks, an attacker tries to exploit the unwanted leakage information from a cryptographic device in the form of timing, power or EM emanations to extract the secret key used for encryption/decryption. Side-channel attacks are difficult to detect since the device under attack is passively observed while operating normally. EM side channel attacks occur when an adversary uses sensitive information from EM signals radiated from cryptographic devices. EM side channel attacks are categorized in two classes: Simple EM attacks (SEMA) and Differential EM attacks (DEMA).

1. Simple EM Attacks (SEMA): Simple EM attacks make use of a single EM voltage trace to visually inspect and look for large scale differences which can correlate the behavior to secret key. If a computation makes use of conditional branches based on secret key, this can be observed on an EM signal during a time interval. E.g. conditional branches dependent on key bit as 0 or 1 will radiate different EM signals and can be deduced by attacker using a single EM trace.

2. Differential EM Attacks (DEMA): In some cases, SEMA does not provide enough information about the secret key and extracting the key would require many more traces. Differential EM attacks require multiple traces and extract secret keys by establishing a statistical relationship between the measured signal and a hypothetical model used to estimate the signal value.

Differential attacks can be broadly categorized into two groups: partition based and comparison based.

1. Partition-based attacks: In a partition-based attack, the measured traces are grouped based on different key hypotheses and then the statistics of these groups are compared. Only the grouping that is done with the correct key guess should reveal a significant statistical difference. For example, assume that the AES EM traces are partitioned into two groups 0 and 1 (for all different key guesses) based on the value of the last bit. For the correct key guess, the group 1 will have signals with a higher

hamming weight on average, thus there will be a EM voltage difference than group 0. For wrong key guesses, the traces will be wrongly partitioned and randomly placed in two bins. Thus, the hamming weights of the two groups will be very similar. There are different techniques based on (1) how to partition the groups and (2) how to extract the EM voltage difference. Difference of Means (DoM) and V-test fall into this category.

2. Comparison-based attacks: In a comparison based attack, the adversary generates a table for each key guesses. A row of this table stores the input message, the corresponding intermediate value for the key guess, a EM voltage model (eg. Hamming weight or Hamming distance) based on this intermediate value, and the actual observed trace. The adversary then tries to find a correlation between the EM voltage model and the actual observed trace. The key guess that has the highest correlation value is selected as the correct key. There are different techniques that are based on (1) how to generate the EM voltage model and (2) how to estimate the correlation. Correlation and Mutual Information based attacks fall into this category.

The generalized flow for a differential attack is summarized in Fig. 2.6. An intermediate state of the algorithm is used to build a hypothesis. Specifically, because AES is byte-based, the hypothesis is based on an 8-bit guessed key with 256 possible values. The model assumes that the EM emission of AES circuit depends on the number of transitions on the state register. The Hamming distance of state registers at the beginning and the end of the final round is used as a predictor of EM emission. This is valid because in CMOS logic the amount of current drawn, and the magnitude of EM field, depends on the number of transitioning gates. If the guessed key is in fact the correct key, the guessed value of Hamming distance for all encryptions matches with the actual number of bit flips. For a wrong key, however, the Hamming distance has a low correlation with the measured trace. The reason is that the SubBytes operation maps every byte to a random byte in each round. This correlation approaches zero as the sample size, i.e., the number of traces T^{EM}, increases. Because the AES transformations are byte-wise, each byte of the secret key can be attacked separately. Thus, differential attacks reduce the search space for the 128-bit secret key from 2^{128} to 2^{12} by attacking one byte at a time [5].

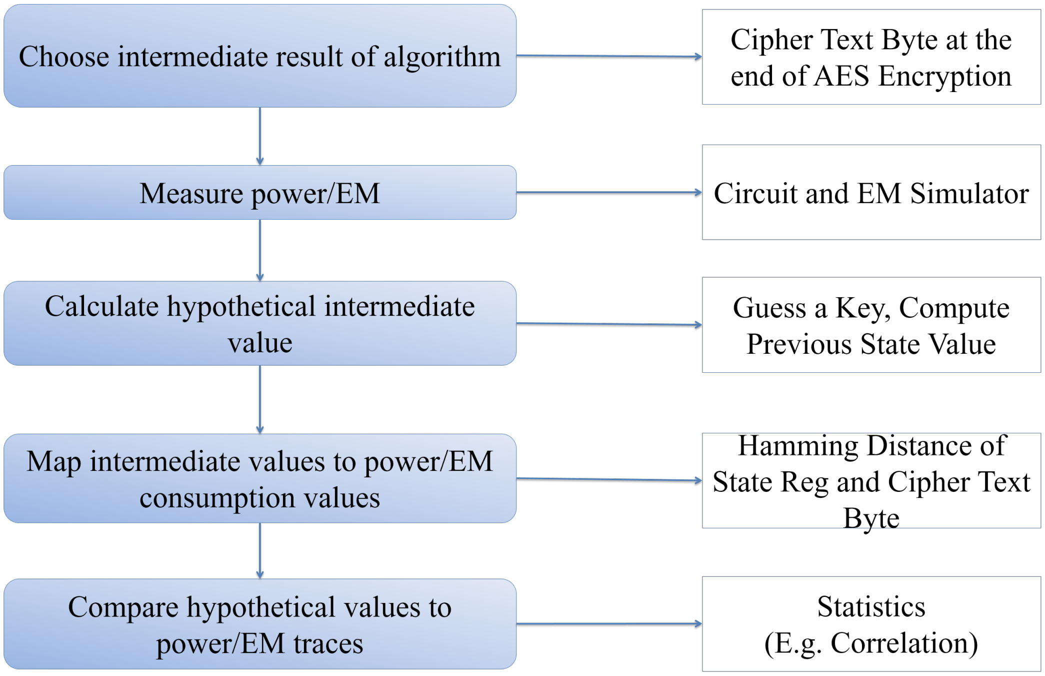

Choose intermediate result of algorithm
Cipher Text Byte at the end of AES Encryption
Measure power/EM
Circuit and EM Simulator
Calculate hypothetical intermediate value
Guess a Key, Compute Previous State Value
Map intermediate values to power/EM consumption values
Hamming Distance of State Reg and Cipher Text Byte
Compare hypothetical values to power/EM traces
Statistics
(E.g. Correlation)

Figure 2.6: Differential Attack Methodology

2.3 Attack metric: Pearson's correlation

The success metric used in this work to evaluate the strength of the attacks is based on Pearson's correlation distinguisher [6]. Pearson correlation checks the linear relationship between two variables based on their covariances. Mathematically, it is defined as:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \times \sigma_X} \tag{2.1}$$

where,

cov is the covariance between X and Y,

σ_X is the standard deviation of X and

σ_Y is the standard deviation of Y

In side-channel attacks, X is a vector containing the hypothetical power model values (e.g. Hamming Distance (HD)) and Y is a vector containing EM voltage values at a particular time instant. This correlation is computed at all time instants, and for all 256 guessed keys. Hence, Pearson's correlation coefficient (ρ) determines a linear relationship between a hypothetical value and measured signal. Specific to AES, the hypothetical power model is the hamming distance. Let's take an example where EM voltages (in mV) measured at a time instant T_1 are 10, 20, 30, 40, 50, 60, 70 and 80. Now, we build power model for each guessed key and computes Pearson's correlation with the measured power and power model. For the correct key, as an example the hamming distances can be 1, 2, 3, 4, 5, 6 and 8 and the linear dependence between the two variables is going to be high resulting in high correlation whereas in the case of wrong guess key, estimated hamming distances could be 5, 8, 1, 3, 2, 7, 4 and 6 leading to low correlation value.

Let's take a case where AES is run for N_e different encryptions and voltage traces corresponding to each encryption are collected, for T time samples. We denote voltage trace corresponding to first encryption as $\{V_1(t_1), V_1(t_2), ..., V_1(t_T)\}$, for second encryption as $\{V_2(t_1), V_2(t_2), ..., V_2(t_T)\}$ and so on...
We build hypothesis for all 256 guess keys and for each guess key, we compute Hamming distance corresponding to each encryption. Let's denote Hamming Distances computed using guessed key 0 as $\{HD_{K_0}(1), HD_{K_0}(2), ..., HD_{K_0}(N_e)\}$,

for guessed key 1 as $\{HD_{K_1}(1), HD_{K_1}(2), ..., HD_{K_1}(N_e)\}$ and for guess key 255 as $\{HD_{K_{255}}(1), HD_{K_{255}}(2), ..., HD_{K_{255}}(N_e)\}$. Let's represent these two variables as matrices.

$$X = \begin{bmatrix} HD_{K_0}(1) & HD_{K_0}(2) & HD_{K_0}(3) & \dots & HD_{K_0}(N_e) \\ HD_{K_1}(1) & HD_{K_1}(2) & HD_{K_1}(3) & \dots & HD_{K_1}(N_e) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ HD_{K_{255}}(1) & HD_{K_{255}}(2) & HD_{K_{255}}(3) & \dots & HD_{K_{255}}(N_e) \end{bmatrix} \tag{2.2}$$

$$Y = \begin{bmatrix} V_1(t_1) & V_1(t_2) & V_1(t_3) & \dots & V_1(t_T) \\ V_2(t_1) & V_2(t_2) & V_2(t_3) & \dots & V_2(t_T) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ V_{N_e}(t_1) & V_{N_e}(t_2) & V_{N_e}(t_3) & \dots & V_{N_e}(t_T) \end{bmatrix} \tag{2.3}$$

Now, Pearson's correlation is computed between each row of X and each column of Y i.e. between hamming distances for each key guess (X) and voltage values at each time instant (Y). Hence, we denote the correlation between row vector $[HD_{K_0}(1), HD_{K_0}(2), HD_{K_0}(3), ..., HD_{K_0}(N_e)]$ and column vector $[V_1(t_1), V_2(t_1), ..., V_{N_e}(t_1)]$ as ρ_{K_0,t_1} and so on ... Thus, at the end we get a correlation matrix as shown below:

$$\rho = \begin{bmatrix} \rho_{K_0,t_1} & \rho_{K_0,t_2} & \rho_{K_0,t_3} & \cdots & \rho_{K_0,t_T} \\ \rho_{K_1,t_1} & \rho_{K_1,t_2} & \rho_{K_1,t_3} & \cdots & \rho_{K_1,t_T} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \rho_{K_{255},t_1} & \rho_{K_{255},t_2} & \rho_{K_{255},t_3} & \cdots & \rho_{K_{255},t_T} \end{bmatrix} \tag{2.4}$$

The coefficients are values from the interval [-1,1] that give an indication about the linear relationship between hypothetical model and measured traces. The key guess corresponding to highest correlation at some time instant is the secret key.

Chapter 3

Related Work

The potency of side-channel attacks, which extract secret keys by exploiting unintended information leakage from physical implementations of cryptographic algorithms with mathematically proven security, has been repeatedly demonstrated over the last two decades [1],[2],[7],[8]. In particular, attacks that analyze the information leaking through power [5] and electromagnetic (EM) [9] side channels present a formidable challenge to ensuring the security of existing cryptographic applications. Effective simulation tools are needed to help designers predict vulnerabilities and improve resilience of cryptographic systems to such SCA attacks. While simulation frameworks that rely on commercial CAD tools to enable design-time prediction of SCA resilience of an implementation to the power side channel are relatively well established [4],[10],[11],[12], only a few attempts have been made at developing them for the EM side channel [13],[14],[15].

Switching activity within the datapath of an implementation during the execution of a cipher can cause information leakage through both the power and EM side channels [5],[9]. Analyzing variations in EM fields received by a probe near the surface of a chip is generally a more effective attack compared to analyzing the chip's total power consumption. This is in large part because the EM channel, which has access to spatially localized emanations, is less influenced by non-cryptographic computations obfuscating the critical activity in crypto-blocks. Indeed, various experiments confirm that EM SCA attacks using near-field sensors retrieve secret keys more efficiently than power SCA attacks [1],[2],[16].

To simulate an EM side-channel attack on a specific IC design, the set of voltage signals ("traces") that will be observed by nearby EM field probes or attached oscilloscope probes during critical cipher execution steps must be computed. This requires modeling across multiple levels of circuit abstraction, transient circuit analysis, and a multitude of similar computations to obtain the many (typically $\sim 10^3 - 10^5$) traces corresponding to the cipher texts that an attacker can observe. Compared to power SCA simulation, EM SCA simulations further require (i) modeling of EM (capacitive, inductive) coupling along the on-chip power/ground/ data/clock interconnects and through the substrate, (ii) transistor-level SPICE analysis to account for information-bearing indirect (modulated) emanations that arise from coupling (e.g., of data signals to clock signals) and the non-linearity of transistor I-V characteristics [9],[13], (iii) linking current/voltage signals found in transistor-level circuit abstraction to layout information, and (iv) using an EM simulator to radiate space-time samples of currents and find the fields that will be received by EM probes. Because of the costs of accurately (i) modeling coupling effects, (ii) analyzing transient signals on transistor-level non-linear circuits, (iii) matching circuit netlists to physical layouts, and (iv) computing EM fields emanated from distributed transient currents, it is significantly more challenging to compute the traces needed for EM SCA attack simulations compared to those needed for power SCA attack simulations.

The computation of each trace used in EM SCA attack simulations can be separated into three steps:

Step 1: Perform a multiple-abstraction-level circuit analysis to determine (time samples of) transient currents on circuit branches during critical cryptographic operations.

Step 2: Extract (space-time samples of) a reduced set of distributed currents on the chip layout from the branch currents found in Step 1.

Step 3: Perform an EM radiation analysis to determine (time samples of) the trace(s) observed by a probe due to currents found in Step 2.

Previous studies [13],[14],[15] use a variety of simplifications to reduce the aforementioned costs of these steps. In [13], the magnetic field radiated near the chip is assumed proportional to the total current and thus the observed voltage (trace) is assumed proportional to the current's time derivative. While this effectively nullifies the costs of Steps 2 and 3, it also reduces the simulations' predictive value; e.g., the simulated trace becomes independent of the probe position. In fact, SCA attack simulations based on this simplification would essentially reduce to predicting vulnerabilities to a power side channel (one that observes the time-derivative of the total current consumption) rather than an EM side channel. In [15], it was proposed to generate switching current models for each cell, represent the currents drawn by the different cells with (impressed) short electric dipoles placed according to the layout of the cells, and find EM fields by radiating these dipoles in free space. This simplifies Step 1 by not modeling the interconnect network and by finding the current drawn by each cell independently; Step 2 by placing a single source for each cell; and Step 3 by radiating point-wise sources. Unlike in [13], the approach in [15] would yield position dependent fields; yet, its predictive value is also expected to be rather limited because it ignores on-chip coupling effects and thus indirect (modulated) emanations. In [14], a commercial dynamic IR drop estimation tool (Apache RedHawk) was used to find signals on the power/ground interconnect network, regularly spaced virtual probes were introduced along the rails to extract the current in the small interconnect pieces between these probes, and the Biot-Savart law was used to find and superpose the magnetic fields from the pieces that are near the EM field probe. This simplifies Step 2 because the extracted currents have specific locations on the interconnect network and Step 3 by using a quasi-static approximation to find EM fields, by only keeping sources on power/ground interconnects, and by discarding emanations from distant sources. Unfortunately, the RedHawk tool used in Step 1, which performs a statistical vector-less analysis of the switching activity to support power delivery network optimization, does not allow event-driven simulation using vector patterns and thus cannot be used to find currents corresponding to different cipher texts observed by an attacker.

This thesis introduces a simulation flow that enables rapid and accurate design-time prediction of EM SCA resilience of cryptographic modules for the first time. Simulation costs are reduced without sacrificing predictive value as follows. Step 1: We use commercial CAD tools to run highly-optimized transistor-level simulations only during critical time windows when information leakage happens. Steps 2-3: We limit the EM radiation to the currents distributed on the top-metallization layer power/ground interconnects and generate traces for different encryptions in parallel. Using the proposed methodology, various differential attacks are simulated on AES block cipher.

Chapter 4

Efficient DEMA Simulation Flow

This chapter presents the proposed design-time simulation flow for evaluating EM SCA attack vulnerability, the details of the three steps (see Chapter 1) used to compute the traces, and the methods used to combat the high costs of accurate simulations without compromising their predictive value. The flow chart in Fig. 4.1 depicts the proposed simulation flow of taking an RTL description of an IC implementation through various simulation steps to extract the secret key.

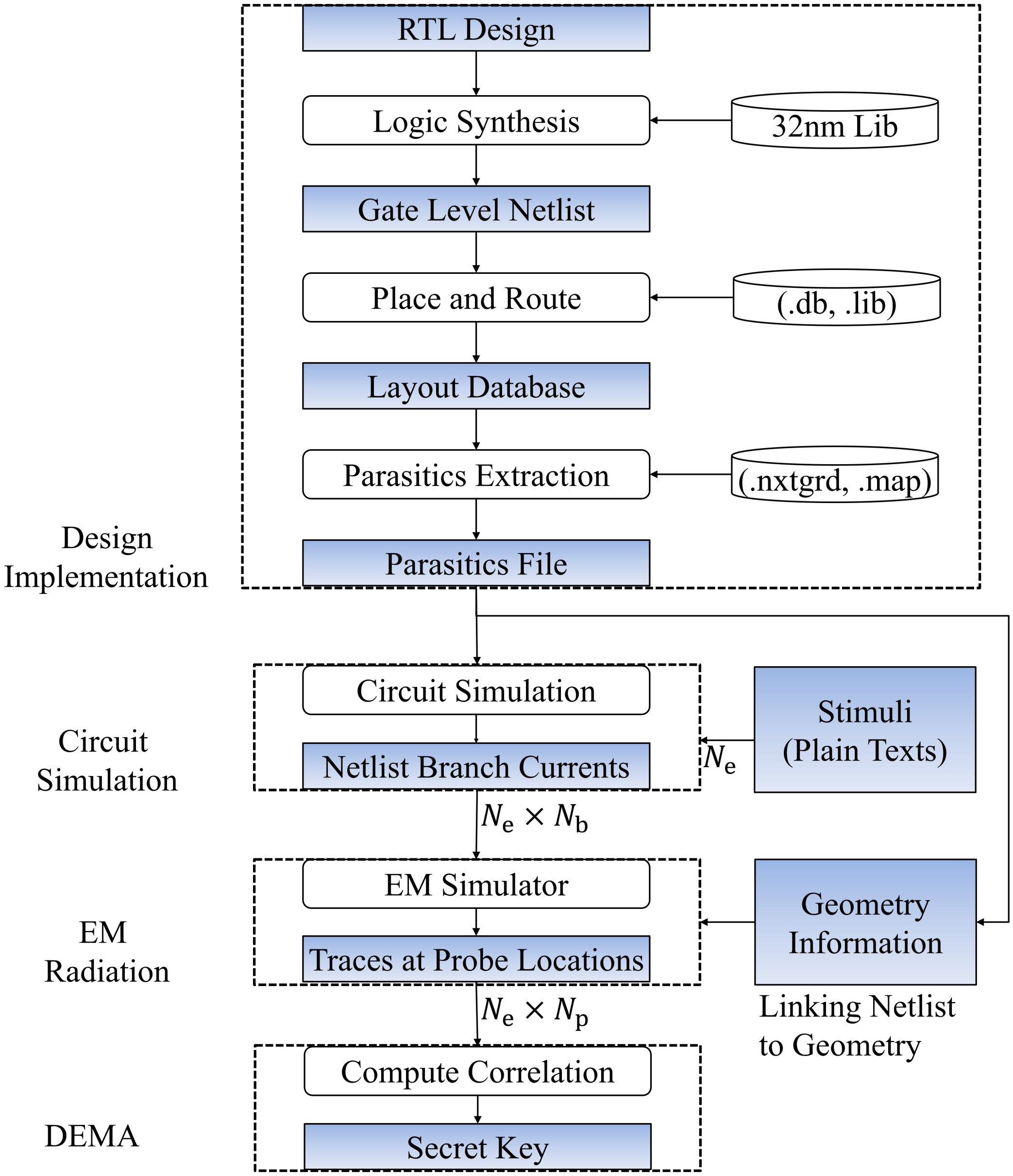


Figure 4.1: Proposed simulation Flow

4.1 Circuit Analysis

Many steps required to produce the traces needed to perform DEMA simulations are obtained directly via standard commercial CAD tools. We reduce runtime significantly by focusing the high-accuracy transient-circuit simulation only on the cipher-execution phase during which the intermediate computations leak information. In the case of AES, this happens in the last round. We propose a hybrid flow of gate-level and transistor-level simulations to take advantage of this fact. Gate-level simulations are performed to compute the state of the circuit at the beginning of the last round and only the critical last round is simulated at transistor level with SPICE.

The simulation flow will be discussed in detail now.

1. **Design Implementation**

    - Logic Synthesis: The RTL level description of a design is synthesized using Synopsys Design Compiler. The implementation is targeted at 32nm technology library. User can specify constraints like target library cells, clock frequency, power and area etc. at this stage. Once these constraints are met, the tool generates a gate level netlist in verilog (.v) format.
    - Place & Route: The synthesized gate-level netlist (.v format) is passed to Synopsys IC Compiler Place & Route tool. The user specifies various parameters like design of power grid, target voltage

drop, metal layers for routing etc.. Once these constraints are met, the database is saved in .mw format.

- Parasitics Extraction: Now, the layout database is used as an input to extract parasitics and generate a spice compatible file in .spf format. Synopsys StarRC tool is used for RC extraction. This step allows the user to dump geometry information of parasitic nodes using EXTRA_GEOMETRY_INFO option during the extraction. Various settings for extraction like extraction of via resistors/capacitors, power-grid parasitic extraction etc. can be utilized at this step. The parasitics file isa saved in .spf format which is compatible with SPICE simulators.

2. **Circuit Simulation**: This steps performs fast SPICE simulations using Synopsys FineSim FastSpice engine. The design file generated in .spf format in the previous step is used as input for SPICE simulations and a stimulus file instantiates the design to perform circuit simulation. The parasitics extraction of the standard cells are taken from reference library files. The user specifies the nodes for which the voltage needs to be printed and the branches for which the current needs to be printed. This step will be discussed in detail in Section 4.2. As discussed earlier, gate-level simulation is performed to generate initial conditions for the SPICE simulation.

   An important challenge in circuit simulation is to identify optimal accuracy level for transistor-level simulations. It is important to realize at this point that the accuracy required for DEMA far exceeds that of power analysis. This is because critical EM signatures come from EM coupling of information-leaking signals to other signals on the chip as well as the fact that the radiation stage computes derivatives of time-varying current signals, a dependence that tends to amplify the impact of small error in current estimation. To identify sufficient/optimal circuit simulation settings, designers must run the attack scenario with different parameters and investigate the convergence of the results as the circuit simulation is made more accurate (see Section 5.2).

The output of the transient circuit analysis is the temporal samples of branch currents during the critical time interval (the last AES round), $t^{start} \leq t \leq t^{end}$. To simplify the EM radiation step, these are generated by constraining the SPICE simulation to uniformly sample the time interval with a time-step size of Δt, i.e., $N_t = \left\lceil (t^{end} - t^{start})/\Delta t \right\rceil + 1$ temporal samples are produced for each branch current at the end of Step 1, where $\lceil x \rceil$ rounds its argument up to the nearest integer.

4.2 Model Simplification

As detailed in Section 4.3, the cost of finding the transient EM radiation scales proportionally to the number of space-time samples of current elements that are radiated; thus, radiating all the currents on all the branches quickly becomes a computational bottleneck as the number of branches in the netlist increases; e.g., the RTL implementation of AES investigated in Chapter 5, has a total of 255728 branches. To limit this cost, in this work, the radiation is computed from a reduced set of currents; specifically, currents only on top metallization layers of the on-chip power-delivery network are radiated. This is a reasonable simplification because the larger dimensions (width, length, thickness) and spacing (pitch) of top-layer interconnects and their proximity to an EM field probe above the chip make them the strongest contributors to the detected signals. As computational capabilities increase, more of the branches can be included in the EM radiation to estimate and reduce the errors introduced by this simplification. Note that a similar simplification was made in [14] and the simulated spatial EM cartographies appeared to be in good agreement with measured data [17].

To identify geometry information for branches corresponding to the top-metallization layer interconnects, the parasitics extractor is used to annotate all the parasitics elements during the CAD flow and to generate a detailed parasitics file in the DSPF format. Then, parsing this file, all but the N_b planar resistive branches at top metallization layers are removed; e.g., for the design in Chapter 5, $N_b = 798$. Next, using the annotations, we attach the coordinates of the four corners of a rectangular patch—a planar surface in 3-D space—and a direction to each of these N_b branch currents. In Step 3, each branch current is assumed to be tangential to and uniformly distributed over the corresponding patch's surface. The patches are assumed to be located at the top surfaces of the metallization layers. Thus, a data file that contains $N_b N_t$ temporal current samples, distributed onto N_b different patches in space, is generated at the end of Step 2.

4.3 EM Radiation

The EM radiation step finds (space-time samples of) the transient fields that would be received by a probe at N_p different positions near the surface of a chip, given the space-time samples of the transient current distribution on the chip. Let $\vec{J}(\vec{r}, t)$ denote the current density at all points $\vec{r}$ on the surfaces S of the (reduced) interconnect network, whose samples are found in Step 2. Then, the magnetic field emanated by these currents in a homogeneous background is given as [18]

$$\vec{H}(\vec{r}, t) = \nabla \times \iint_S \frac{\vec{J}(\vec{r}', t - R/c)}{4\pi r} ds' \tag{4.1}$$

$$\vec{H}(\vec{r}, t) = \frac{1}{4\pi} \iint_S ds' \hat{R} \times \left[\frac{1}{cR} \frac{\partial \vec{J}(\vec{r}, t')}{\partial t'} + \frac{\vec{J}(\vec{r}, t')}{R^2} \right]_{t' = t - \frac{R}{c}} \tag{4.2}$$

Here, c is the speed of light in the background medium, R and $\hat{R}$ are the magnitude and direction of the vector $\vec{R} = \vec{r} - \vec{r}'$ that is directed from the source point at $\vec{r}'$ to the observer point $\vec{r}$. In the near field of the source, the $\vec{H}$ field is determined mainly by the second term on the right-hand side, which decreases quadratically with distance—this is the physical basis for the simplification proposed in Step 2. Assuming the field is received by a probe that is a single-turn loop of wire encircling the surface S_p, which is centered at position $\vec{r_p}$ and has three possible orthogonal orientations identified by the loop's normal direction $\hat{u} \in \{\hat{x}, \hat{y}, \hat{z}\}$, the voltage signal (an EM trace) detected at the loop terminals (determined by the magnetic flux through the loop's surface according to Faraday's law) is given as:

$$V_p^u(t) = -\frac{d}{dt} \iint_{S_p} \mu \vec{H}(\vec{r}, t) \cdot \hat{\mu} ds \tag{4.3}$$

where μ is the magnetic permeability of the loop core. Clearly, the contribution of a current element to the trace received by the probe depends on its direction relative to the probe orientation as well as its distance to the probe. If the N_p different positions of the probe are simulated over the chip surface to obtain a spatial cartography, a total of $3N_p$ EM traces $(V_1^{x,y,z}, ..., V_{N_p}^{x,y,z})$ are generated.

To numerically calculate the detected voltages, $\vec{J}(\vec{r}, t)$ is discretized using space-time basis functions as:

$$\vec{J}(\vec{r}, t) \approx \sum_{k=1}^{N_b} \sum_{l'=1}^{N_t} I_{k,l'} \vec{S_k}(\vec{r}) T_{l'}(t) \tag{4.4}$$

Here, N_t temporal basis functions $T_1, ..., T_{N_t}$ and N_b spatial basis functions $\vec{S_1}, ..., \vec{S_{N_b}}$ are used to approximate the current density. As is common, piecewise polynomial (sub-domain) interpolatory functions are used as temporal basis functions. The spatial basis functions are constant (vector) pulse functions on rectangular patch surfaces that are parallel to one side of the patch and normal to the other. Thus, each expansion coefficient corresponds one-to-one to a current sample extracted in Step 2, scaled by a constant equal to the width of $S_{k'}$.

Once the current discretization is substituted into the above integrals, the EM trace at each probe position p, orientation $\hat{u}$, and time t_l is given as

$$V_p^u(t_l) = \sum_{k=1}^{N_b} \sum_{l'=1}^{N_t} Z_{p,l,k,l'}^u I_{k,l'} \tag{4.5}$$

where

$$Z_{p,l,k,l'} = -\frac{\mu}{4\pi} \iint_{S_p} \hat{\mu} \cdot \iint_{S_k} \hat{R} \times \vec{S}_k(\vec{r}) \left[\frac{\ddot{T}_{l'}(t_l - \frac{R}{c})}{cR} + \frac{\dot{T}_{l'}(t_l - \frac{R}{c})}{R^2} \right] ds' ds \tag{4.6}$$

$\dot{T}_{l'}$ and $\ddot{T}_{l'}$ are the first and second derivatives of the temporal basis functions and the inner integrals are over the patch k surface. To compute the above integrals, numerical quadrature rules with $N_{q,s}$ points on each source patch and $N_{q,o}^u$ points over the observer surfaces is used. Let $N_q = (N_{q,o}^x + N_{q,o}^y + N_{q,o}^z) X N_{q,s}$, then filling all $Z_{p,l,k,l'}^u$ entries requires $O(N_p N_b N_q)$ operations as the interactions are sparse and most $Z_{p,l,k,l'}^u$ are zero. Still, these are expensive computations (with a large constant in front) that should be performed once and amortized over all N_e encryptions. To simplify the book keeping, a uniform time discretization is assumed for the currents, i.e., $T_l(t) = T(t - l\Delta t)$, and the received voltages are assumed to be recorded at the same time instances as the currents. Other algorithms [18] can also be adopted as $N_p N_b$ increases.

Once $Z^u_{p,l,k,l'}$ are filled and stored, computing the $3N_p$ EM traces requires $O(3N_pN_bN_t)$ floating-point multiply-add operations for each encryption.

4.4 Computation Costs and Parallelization

To simulate DEMA, the circuit analysis and EM radiation steps are performed N_e times, each time corresponds to a different encryption that can be observed by an attacker. To reduce simulation times, we propose to generate the EM traces corresponding to each encryption in parallel. The circuit analysis computations are assumed to be distributed among P^{CKT} processes, which may be limited by the number of available licenses when commercial CAD tools are used. The custom EM radiation simulations are assumed to be distributed among P^{EM} processes. Table 4.1 summarizes the dominant computational costs for performing a DEMA simulation for N_e encryptions. In the table, t^{CKT}_{1enc} denotes the circuit analysis time needed for 1 encryption and t^{EM}_{core} denotes the time needed for finding the contribution of each branch to the voltage signal for 1 probe positionand 1 encryption. Table 4.1 also shows the speedups we observed when using the proposed methodology for the simulations in Chapter 5. Here, using HSPICE for circuit simulation at all rounds of the AES is denoted as the naïve approach and radiating all branch currents in the EM simulation is denoted the brute-force approach.

Simulation Step	Dominant Cost (Wall-clock time)	Observed Simulation Times
Circuit Analysis	$t_{1enc}^{CKT} \times \left\lceil \frac{N_e}{P^{CKT}} \right\rceil$	$t_{1enc}^{CKT} \approx 20hr(naive)$ $t_{1enc}^{CKT} \approx 0.1hr(proposed)$ Sample Attack: $N_e = 5000$ $P^{CKT} = 30$ $total \approx 20hr$
EM Radiation Analysis	$t_{core}^{EM} \times N_b \times N_p \times \left\lceil \frac{N_e}{P^{EM}} \right\rceil$	$t_{core}^{EM} = 12\mu s$ $N_b = 255728(bruteforce)$ $N_b = 798(reducedmodel)$ Sample Attack: $N_e = 5000, N_p = 36$ $P^{EM} = 1$ $total \approx 0.5hr$

Table 4.1: Simulation Times for computing an attack on N_e encryptions using the proposed flow

Chapter 5

Results

In this chapter, the proposed simulation flow is evaluated by applying it to various EM SCA attack vulnerability analysis of an ASIC implementation of AES.

5.1 System Setup

The RTL implementation of AES is from [19] . The design is implemented using 32 nm CMOS technology library and occupies a $230\mu\text{m} \times 230\mu\text{m}$ region of the chip. The interconnect networks were placed using orthogonal (HVH) routing and the P/G networks were routed in the top two metal layers M7 and M8. The design is supplied by 8 VDD and 8 VSS sources present at the ends of straps and symmetrically distributed around the ring in M8 to keep the IR-drop in check. We use the circuit simulation methodology described in Section 4.1 to generate branch currents typically for $N_e = 5000$ different encryptions. Gate-level simulations were performed for all but the last AES round using the VCS tool and transistor-level simulations were performed for the last round using the FineSim tool from Synopsis. A time resolution of $\Delta t = 10ps$ was used in transient circuit analysis, which resulted in $N_t = 1501$

time samples in the last round of the AES cycle. Only $N_b = 798$ branch currents that correspond to the P/G network in the top two metal layers M7 and M8 were passed on to the EM radiation step. Importantly, the wires in M7, which are along the x-direction generate EM radiation in a different orientation compared to the emission from M8 wires, which are along y-direction.

In EM simulations, the probe was modeled as a square loop of 25μm side length. Consistent with the behavior of the physical probes used in [2],[14],[17], the signal at the output of the probe was processed using a low-pass 5^{th} order Chebyshev filter with a 1-GHz cut-off frequency and 0.002-dB passband ripple. The probe was located at different positions over the chip, in three orientations, and its distance from the top surface of M8 layer was varied from 10-μm to 1-mm to observe how attack results change. To demonstrate the flow, we first perform a spatial cartography by moving a y-oriented loop placed, whose center was placed 30 μm above the chip with a displacement step size of 46 μm. The results are produced using 1 transmitter quadrature point and 1 receiver quadrature point. Fig. 5.1 shows the strength of EM signal recorded at time step 500.

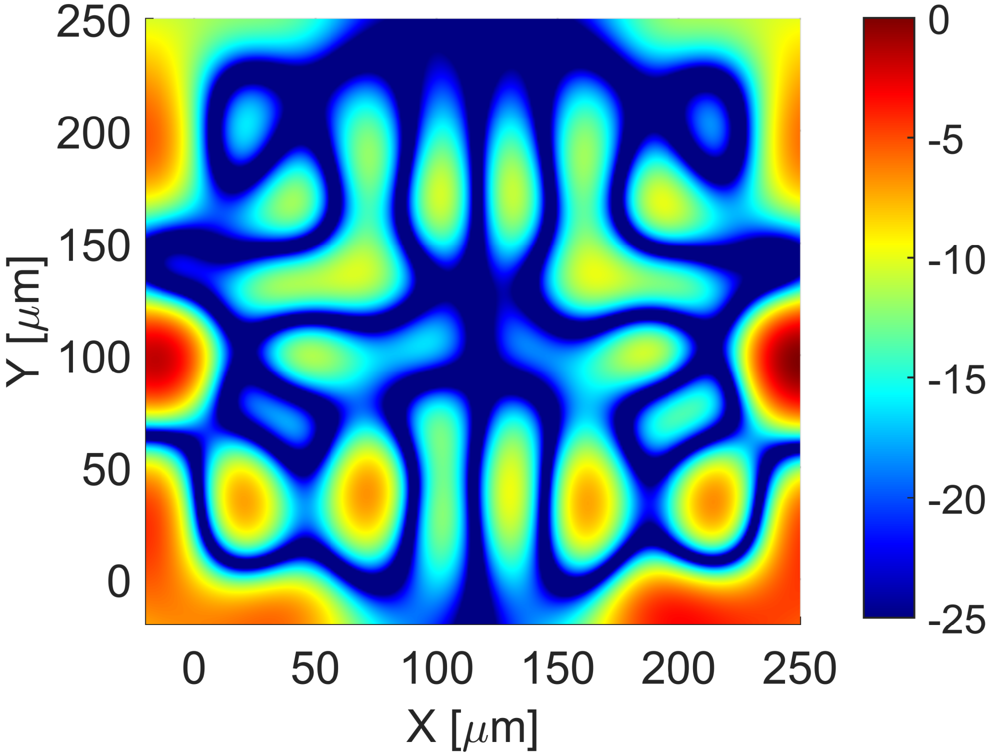
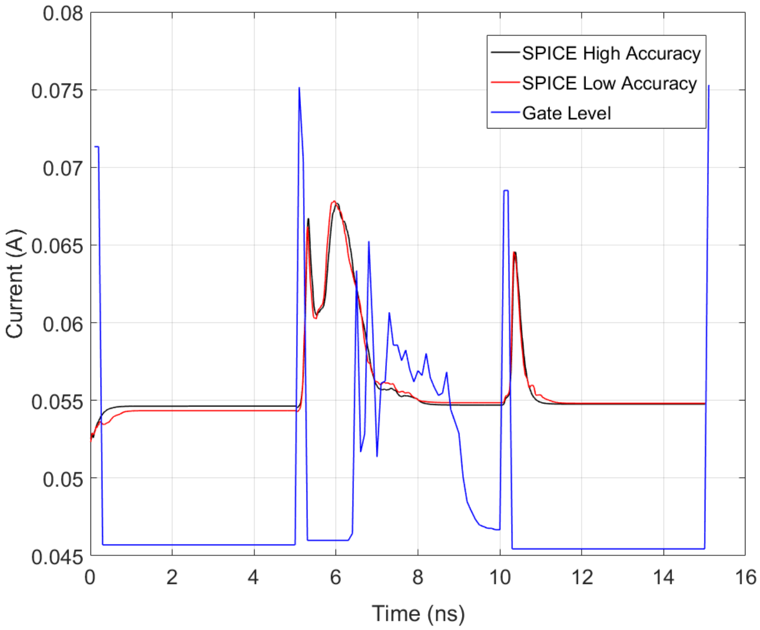


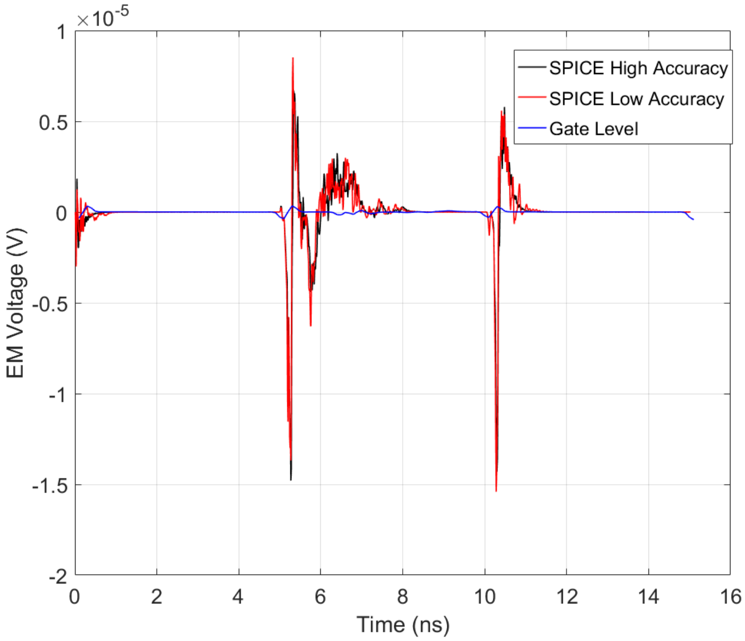
Figure 5.1: EM spatial cartography above the AES block. The detected voltages are shown at a time instant corresponding to the rising edge of the clock at the end of the last round of AES.

5.2 Circuit Analysis Accuracy

We evaluate the accuracy needed in circuit simulations by computing the confidence ratio as a function of number of traces in a DPA attack and a DEMA attack. The confidence ratio is a simple metric of attack success and is defined as the ratio of the highest and the second-highest differential peaks among all 256 key guesses. In the following experiments, the EM traces



(a) Current waveforms



(b) EM Voltage waveforms

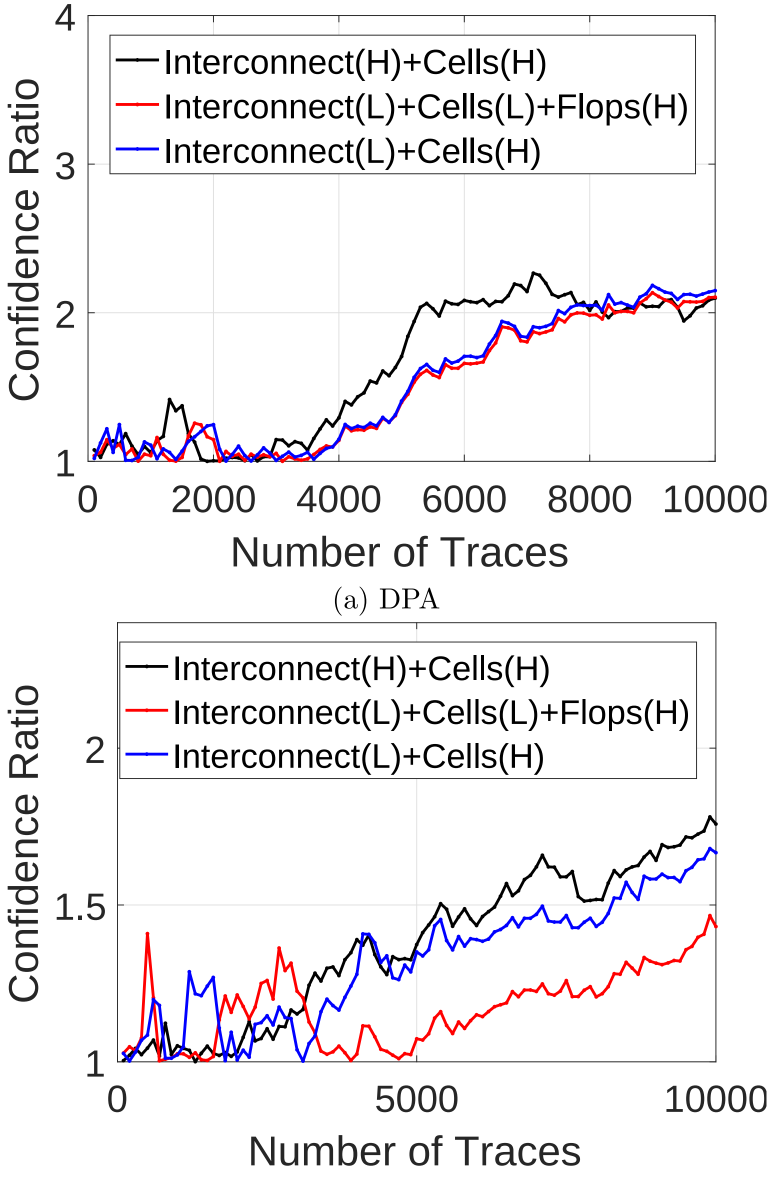
Figure 5.2: Effect of accuracy level on current and EM voltage traces.

were generated using the y-oriented loop at a single position 500 μm above the center of the chip, i.e., the number of EM traces generated is the same as the number of encryptions.

The first task is to identify the level of accuracy required for predictive simulation. To realize this, we look at the current and EM voltage traces for three different simulations: tx-level high accuracy, tx-level low accuracy and gate-level simulations. It is clearly seen that gate level simulation is not as accurate as tx-level as shown in Fig. 5.2a. Further, looking at EM voltage traces, we say that low accuracy tx-level might be sufficient for DPA, but not for DEMA since small differences in current waveforms lead to derivative terms in EM voltage waveforms which lead to big differences as visible in Fig. 5.2b.

First, we look at the effect of parasitic extraction granularity on the attack and coarsen the minimum resolution of parasitic elements in the simulation from $1m\Omega$ to 1Ω , which speeds up the circuit simulation due to fewer parasitic nodes. Fig. 5.4 shows that this does not impact the simulation of DPA attack and designers can choose to go with higher granularity values for extraction to avail speedup. However, this trend does not hold true in case of EM analysis attack.

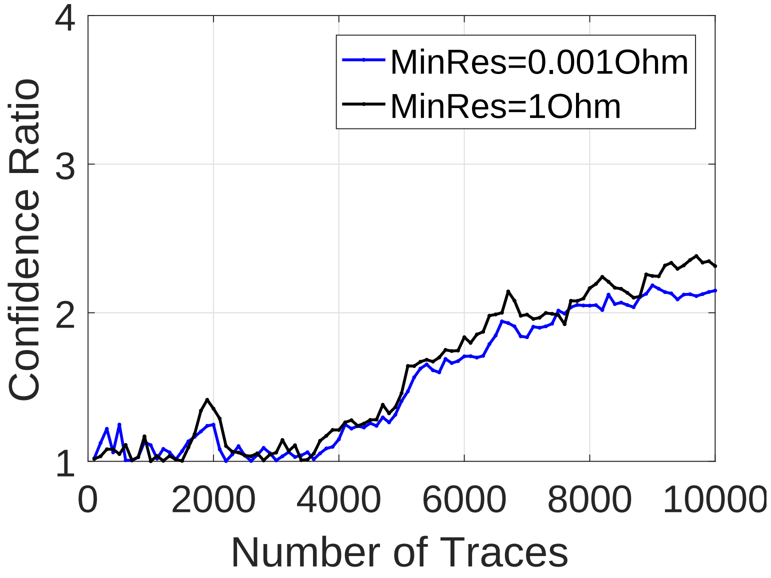
Another crucial optimization parameter for the analysis is the acceptable accuracy level for various circuit components. Fig. 5.3 shows the confidence ratio for differential analysis when high (H) or low (L) accuracy settings are used for three different components. The results indicate that simulating standard cells at the relaxed accuracy affects DPA results much less compared



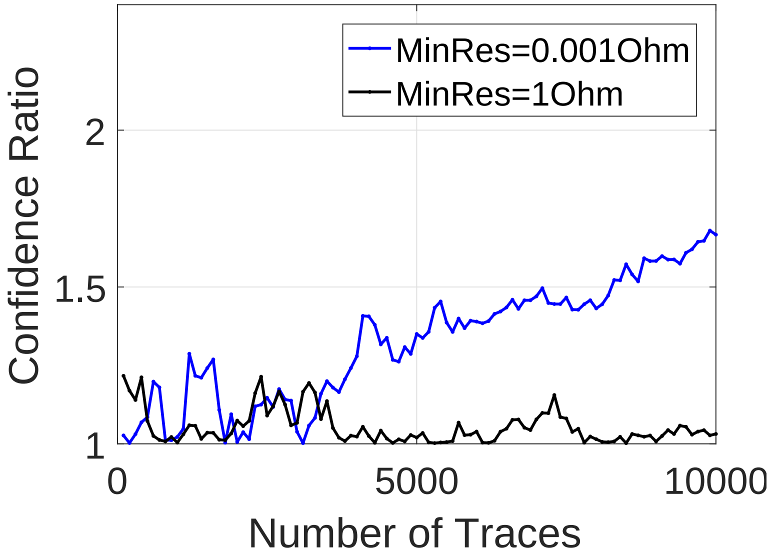
(a) DPA

(b) DEMA

Figure 5.3: Dependence of DPA and DEMA's convergence on simulation accuracy.



(a) DPA



(b) DEMA

Figure 5.4: Dependence of DPA and DEMA's convergence on minimum resistance.

MINRES	Simulation Time		

	Interconnect(H)
	+Cells(H)
		Interconnect(L)
		+Cells(L)+Flops(H)
			Interconnect(L)
			+Cells(H)

0.001	723	391	401
0.01	715	385	397
0.1	637	217	232

Table 5.1: Simulation Time vs Accuracy Parameters

to DEMA ones, e.g. the red curves in Fig. 5.3 that correspond to the scenario where only flip-flops are simulated at SPICE accuracy is not sufficient for EM analysis but acceptable for power analysis. Based on these results, we conclude that all standard cells must be simulated at SPICE level accuracy to accurately predict DEMA attacks. Speed up is achieved by simulating interconnect at lower accuracy and by utilizing FineSim's muti-core/multi-machine simulation capability. This is especially useful while generating data to perform differential attacks, which can require several thousands of traces.

From Table 5.1, it is evident that using high accuracy mode for interconnect is much more expensive than using high accuracy mode for standard cells. This is, however, due to an algorithmic difference rather than an indication of nodes in the interconnect. Also, going down to 1Ω parasitic extraction does not lead to a significant reduction simulation time, but hampers the simulation accuracy a lot.

5.3 Spatial EM Attacks

This section demonstrates the capabilities that the proposed simulation flow provides to designers. We perform various DEMA experiments and compare the results at different spatial locations for the probe. First, we show how a probe can be moved to identify the maximally correlating areas. Next, we show that this helps in identifying trade-offs during physical implementation of such ciphers.

The success metric used to evaluate the strength of the attacks is based on Pearson correlation distinguisher [6]. Pearson correlation coefficient(ρ) determines a linear relationship between a hypothetical value and measured signal. The metric was discussed in Section 2.3. To determine the number of traces required to break the secret key, we use a null hypothesis as ρ=0 and a confidence level of 99.999% to reject the hypothesis. Rejecting the null hypothesis leads to a successful attack. To demonstrate the use of this success metric, we simulate a DEMA attack using the y-oriented loop at a single position 10 μm above the chip near its center; the number of EM traces is again the same as the number of encryptions. Fig. 5.5 depicts a successful attack where we plot ρ for all 256 guessed keys with 5000 traces. Black dotted line represents the 99.999% confidence for the null hypothesis. The figure shows one of the guessed keys (shown in black) rejects the null hypothesis and crosses the confidence level, leading to a successful attack after $\sim$ 1000 traces.

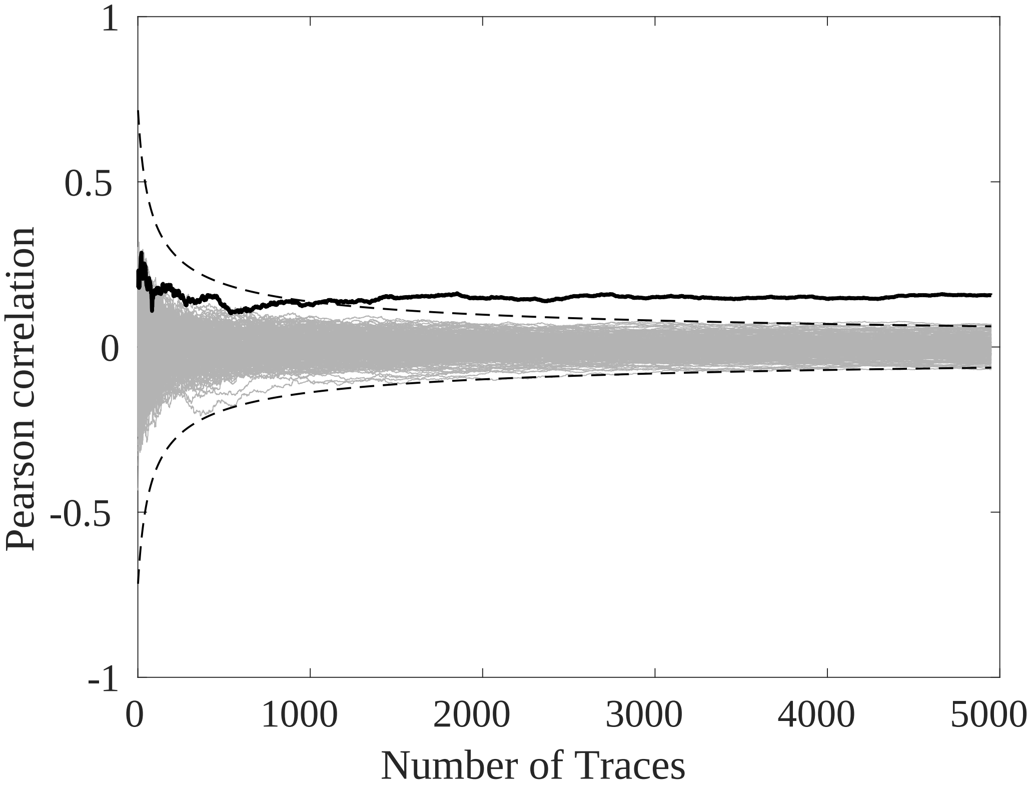
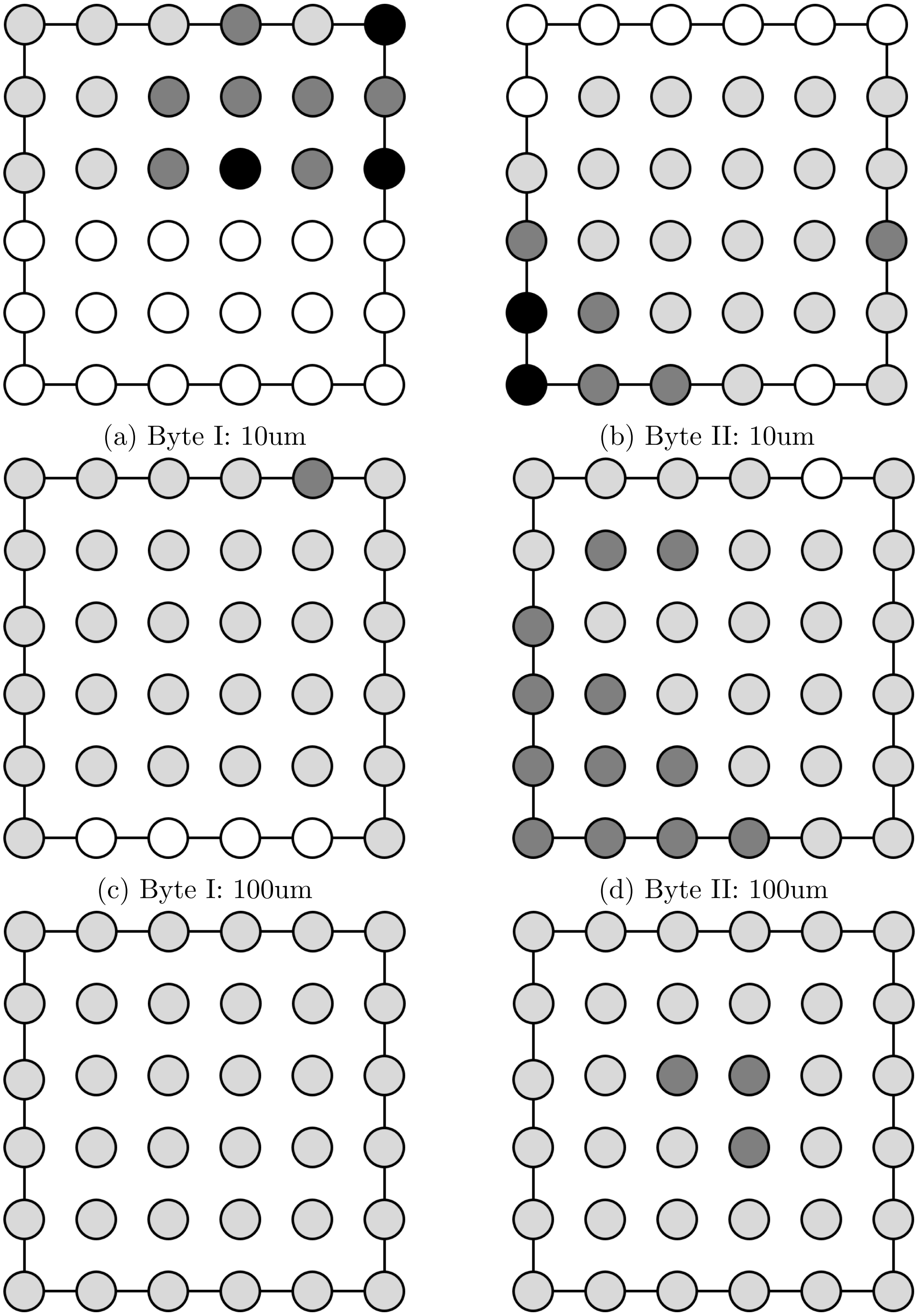


Figure 5.5: Evolution of Pearson correlation coefficient with number of EM traces.

Next, we perform spatial cartography using a probe whose center is at 10μm, 100μm, and 1000μm from the chip surface and identify spots where high correlation is observed. Fig. 5.6 shows this trend when the attack is repeated at 36 probe positions. The critical revelation is that at closer distances to the chip, the points of successful attack (the darker spots in Fig. 5.6) are localized. Moreover, the attack requires much smaller number of traces to succeed when the probe is closer to the chip surface (see Table 5.2).



(a) Byte I: 10um

(b) Byte II: 10um

(c) Byte I: 100um

(d) Byte II: 100um

(e) Byte I: 1000um

(f) Byte II: 1000um

Figure 5.6: Spatial attack on two different bytes of secret key.

To further explore this dependence, a different byte of the secret key was attacked and the analysis was repeated as shown in Fig. 5.6b, 5.6d and 5.6f. It was found that placing the probe closer to the chip still leads to higher correlation values. The positions corresponding to successful attacks shift to different locations in space, as opposed to the case where the probe is farther from the chip when the positions of success stay roughly the same.

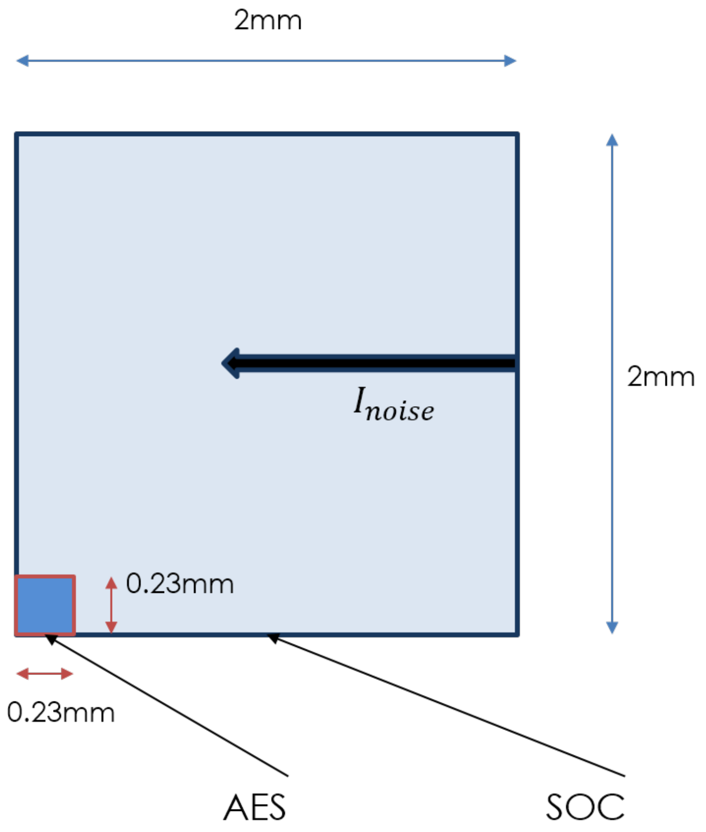
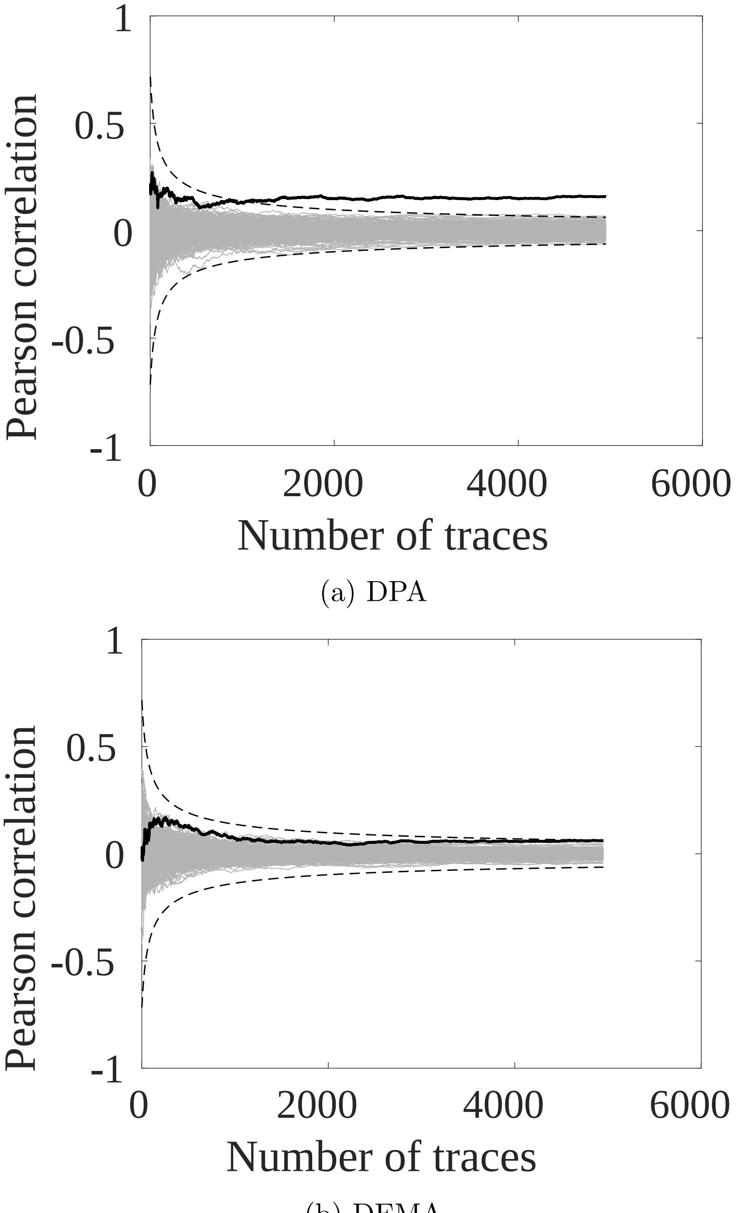


Figure 5.7: Noise source added to AES chip

The effect of proximity of EM probe was further studied by recording the number of traces for a successful attack as the probe height is modified under noisy conditions. To show how DEMA performs under the influence of noise, we introduce a Gaussian noise source to model the impact of the rest of the SOC. The noise source is modeled with a mean current of 1000 mA and a standard deviation of $\sigma = 0.5\%$ of the mean. The dimensions of the SOC are assumed to be $2 \times 2mm^2$, whereas the AES dimensions are $0.23 \times 0.23mm^2$ as depicted in Fig. 5.7. Fig. 5.9 summarizes the findings and shows the effect of noise sources on DEMA and DPA. The blue threshold lines in Fig. 5.9 represent the number of traces required for a successful DPA attack on AES. Fig. 5.8 shows one such instance where DPA fails with even 5000 traces; notice that DPA was successful with only 300 traces when no other source was modeled. In contrast, Fig. 5.9 shows that EM attack results did not get affected as much, since the EM signals from the noise source decay rapidly within a few μm from the surface of the chip when we are at closer distances. Indeed, when the probe is placed $10\mu m$ above the chip, the number of traces required to succeed increased meagerly from 1025 to 1030 when the noise source was modeled. This confirms that DEMA can isolate information leaking circuit elements from the rest of the chip. Fig. 5.9 shows that as the distance of the probe from the chip increases, however, the interference from the noise source starts to affect DEMA performance.



(a) DPA

(b) DEMA

Figure 5.8: Effect of noise on DPA and DEMA attacks when $\sigma = 0.5\%$.

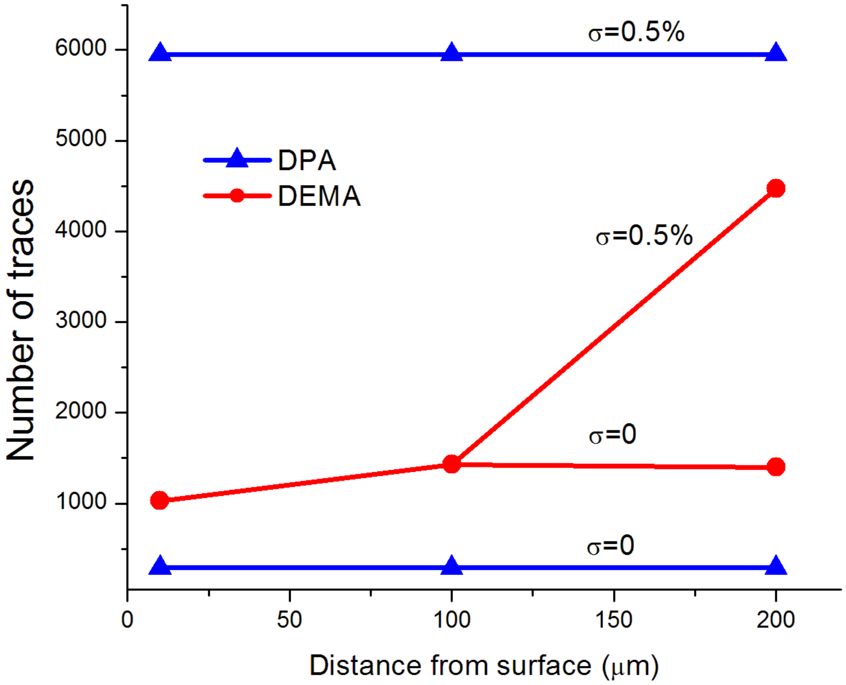


Figure 5.9: Effect of noise on DEMA and DPA

Next we show how early-stage planning is essential and how designers can make critical design choices while implementing the cipher. One of the aspects that is not critical for evaluating power attack resilience is the on-chip power distribution network. This design feature becomes critical, however, for determining EM attack vulnerability as the EM emanations of the hardware depends heavily on it. To study this impact, we simulate DEMA attacks on two different physical implementations of AES. In the first case, the design is implemented (possibly to satisfy routing constraints) by using uniformly spaced VDD and VSS metal lines, which is expected to cause significant EM emanations. The actual layout of power grid network is shown in Fig. 5.11. This is achieved using the following command: *set_fp_rail_constraints -spacing interleaving*

In the second case, the layout is done by placing the two metal lines close to each other, which is expected to reduce the emanations, especially farther away from the chip. This layout is shown in Fig. 5.10. We perform DEMA attacks over 36 spatial points and record the least number of traces required to break AES under noise-free conditions; the results are shown in Table 5.2, which confirms that a uniformly spaced P/G network is a poor design choice that can lead to a highly vulnerable implementation of AES. Indeed, when VSS and VDD segments are close, the overall signal strength is reduced leading to a higher number of traces required to break the key.

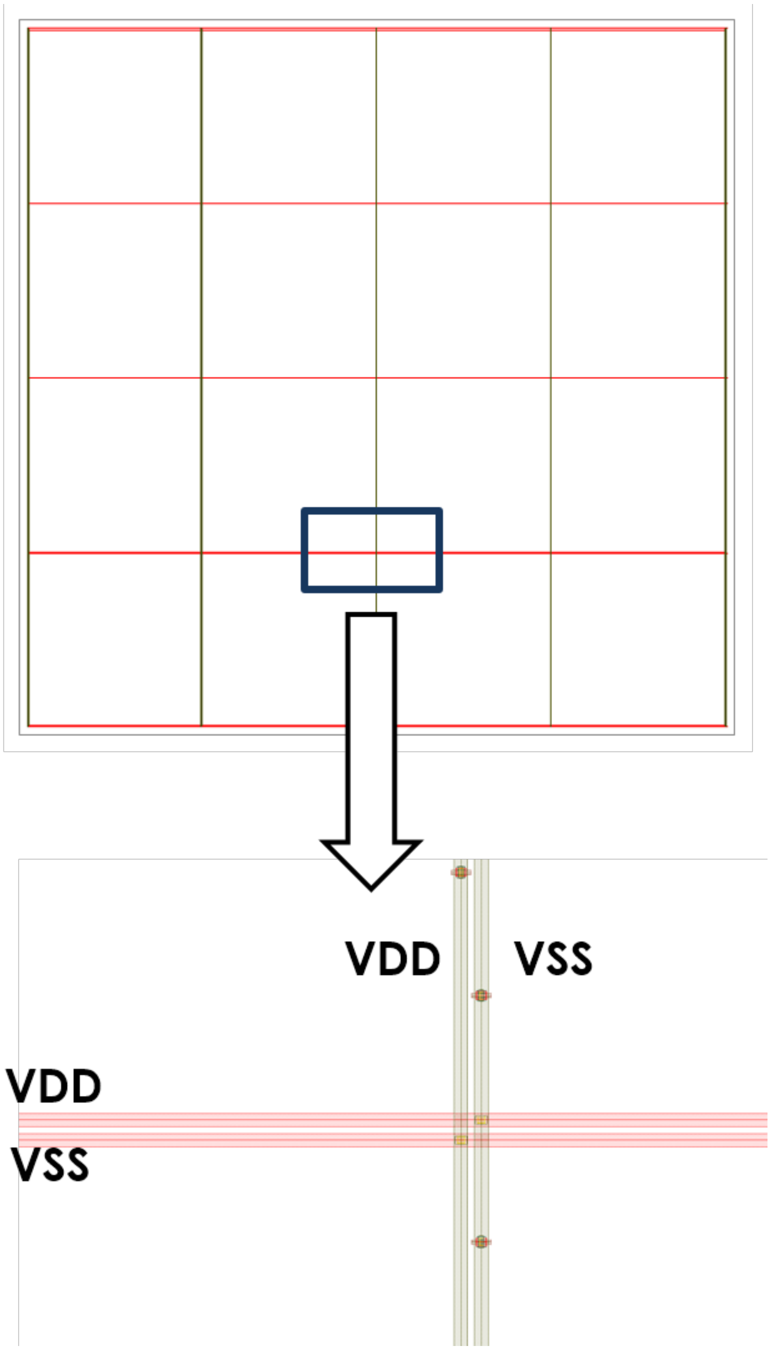


Figure 5.10: Non-uniformly spaced P/G network

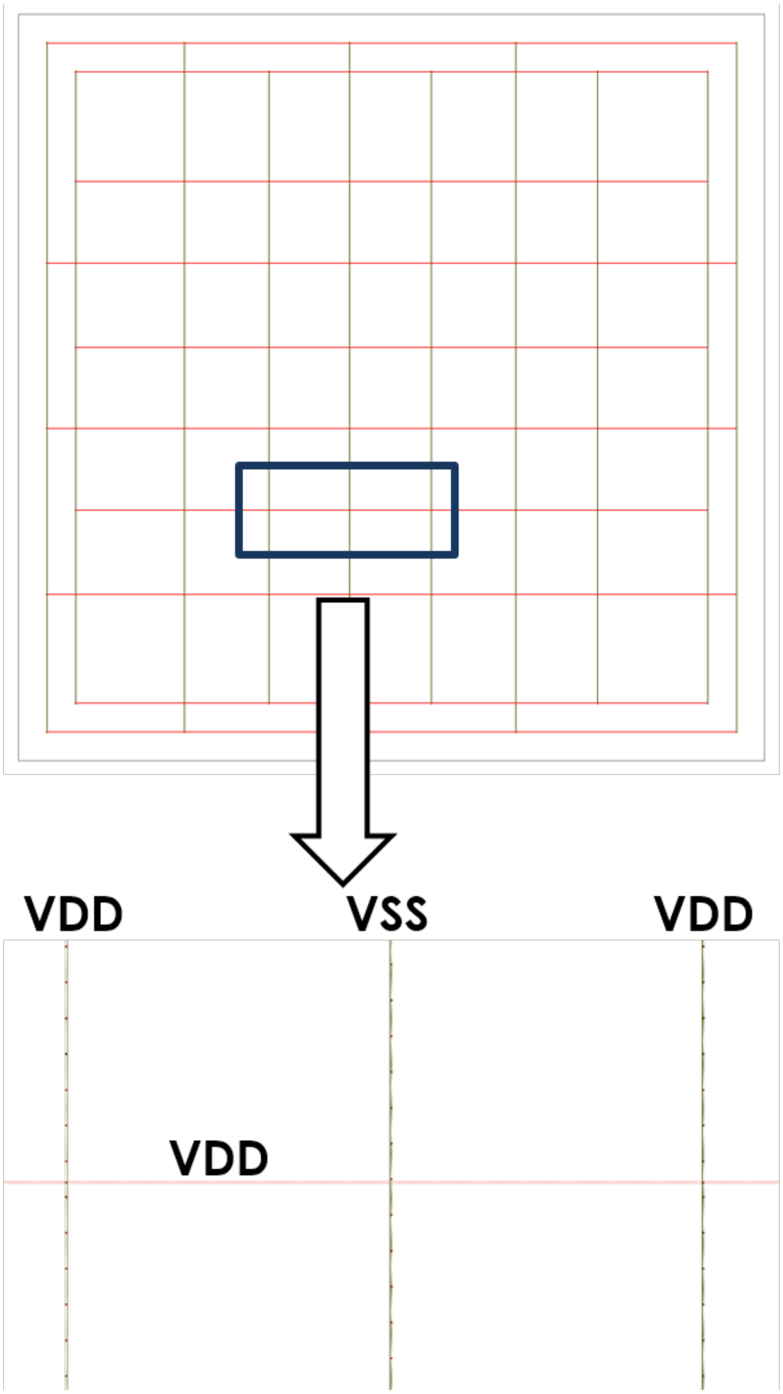


Figure 5.11: Uniformly spaced P/G network

	Number of Traces	
Probe Distance from chip (μm)	Non-uniform spacing	Uniform spacing
10	85	76
100	950	240
1000	1470	310

Table 5.2: Traces required for a successful attack on two implementations of AES

Chapter 6

Summary & Limitations

This chapter summarizes the work, addresses limitations of current research and possible future improvements.

6.1 Summary

This thesis presented a methodology for accurately predicting vulnerabilities of crypto-systems against EM side-channels at design time. The proposed method employs industry-standard CAD tools to extract critical information required to generate EM signatures. This enables designers to simulate differential EM attacks by considering full chip parasitic elements for the first time. The work also demonstrated the benefits of monitoring EM channels as compared to power because of its localization effects. It was shown that certain optimizations in the simulation flow allow designers to assess security in a highly efficient manner. With the preliminary analysis and simple modeling techniques, it has been shown that points of success are more localized at closer distances to the chip and attack becomes successful with fewer traces. This automated design flow allows designers to replace standard CMOS libraries with protected logic and validate the security features of the hardware block.

6.2 Limitations & Future Improvements

This approach is the first step towards building a predictive verification environment for studying simulated EM attacks. Now, we will discuss about the limitations of current work.

1. Limited quadrature points: The results presented in this work were obtained using 1 transmitter quadrature-point and 1 receiver quadrature-point. A convergence study needs to be done to find out the optimal number of source and receiver quadrature points for small and big probe sizes. The results presented in this work are subject to change when more quadrature points are used for EM radiation.

2. Modeling more metal layers: Current work focuses on radiating from top two metal layers of power grid interconnect and neglects signals/power grid parasitics present in lower metal layers with the assumption that top metal layers contribute the most to EM radiation. Further studies need to be done in this regard. This would require faster algorithms to be adopted for EM simulations.

3. Modeling VIA resistors: We did not include via parasitics for EM radiation and since the orientation of vias is different from the other parasitics elements present in with M7 or M8, these should be included to study the effect.

4. Shielding/packaging: Further, the effect of shielding and packaging could be considered and the effect of these factors should be observed.

5. Predictability of results: Finally, real silicon experiments should be performed to validate the simulations and to stdy the predictability of results using the proposed simulation tool flow.

6. Unprotected design: Currently, the attack results are shown for an unprotected design of AES cipher which requires few hundred thousands traces. We sould consider protected implementations of ciphers which would require several hundred thousand traces. Foor such analysis, further speedup needs to be investigated in circuit and EM simulations.

Bibliography

[1] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, "Power and electromagnetic analysis: Improved model, consequences and comparisons," *Integration, the VLSI journal*, vol. 40, no. 1, pp. 52–60, 2007.

[2] L. Sauvage, S. Guilley, and Y. Mathieu, "Electromagnetic radiations of fpgas: High spatial resolution cartography and attack on a cryptographic module," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 2, no. 1, p. 4, 2009.

[3] N.-F. Standard, "Announcing the advanced encryption standard (aes)," *Federal Information Processing Standards Publication*, vol. 197, pp. 1–51, 2001.

[4] S. B. Ors, F. Gurkaynak, E. Oswald, and B. Preneel, "Power-analysis attack on an asic aes implementation," in *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 2. IEEE, 2004, pp. 546–552.

[5] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in cryptology—CRYPTO'99.* Springer, 1999, pp. 789–789.

[6] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware*

and Embedded Systems. Springer, 2004, pp. 16–29.

[7] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, “Ecdh key-extraction via low-bandwidth electromagnetic attacks on pcs,” in *Cryptographers’ Track at the RSA Conference.* Springer, 2016, pp. 219–235.

[8] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, “Ecdsa key extraction from mobile devices via nonintrusive physical side channels,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* ACM, 2016, pp. 1626–1638.

[9] D. Agrawal and B. Archambeault, “Rao and jr, rohatgi, p.:“the em side-channel (s): Attacks and assessment methodologies”,” *Cryptographic Hardware and Embedded Systems–CHES*, 2002.

[10] K. Tiri and I. Verbauwhede, “A vlsi design flow for secure side-channel attack resistant ics,” in *Proceedings of the conference on Design, Automation and Test in Europe-Volume 3.* IEEE Computer Society, 2005, pp. 58–63.

[11] F. Regazzoni, S. Badel, T. Eisenbarth, J. Grobschadl, A. Poschmann, Z. Toprak, M. Macchetti, L. Pozzi, C. Paar, Y. Leblebici *et al.*, “A simulation-based methodology for evaluating the dpa-resistance of cryptographic functional units with application to cmos and mcml technologies,” in *Embedded Computer Systems: Architectures, Modeling and Simulation, 2007. IC-SAMOS 2007. International Conference on.* IEEE, 2007, pp. 209–214.

[12] K. Smith and M. Łukowiak, "Methodology for simulated power analysis attacks on aes," in *MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010.* IEEE, 2010, pp. 1292–1297.

[13] H. Li, A. T. Markettos, and S. Moore, "Security evaluation against electromagnetic analysis at design time," in *International Workshop on Cryptographic Hardware and Embedded Systems.* Springer, 2005, pp. 280–292.

[14] V. Lomné, P. Maurine, L. Torres, T. Ordas, M. Lisart, and J. Toublanc, "Modeling time domain magnetic emissions of ics," in *International Workshop on Power and Timing Modeling, Optimization and Simulation.* Springer, 2010, pp. 238–249.

[15] M. Yoshikawa and T. Asai, "Platform for verification of electromagnetic analysis attacks against cryptographic circuits," in *Information Technology: New Generations (ITNG), 2013 Tenth International Conference on.* IEEE, 2013, pp. 653–658.

[16] 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.

[17] T. Ordas, M. Lisart, E. Sicard, P. Maurine, and L. Torres, "Near-field mapping system to scan in time domain the magnetic emissions of integrated circuits," in *International Workshop on Power and Timing Modeling, Optimization and Simulation.* Springer, 2008, pp. 229–236.

[18] A. E. Yilmaz, J.-M. Jin, and E. Michielssen, "Time domain adaptive integral method for surface integral equations," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 10, pp. 2692–2708, 2004.

[19] A. Satoh, http://www.aoki.ecei.tohoku.ac.jp/crypto/.

[20] F. Regazzoni, A. Cevrero, F.-X. Standaert, S. Badel, T. Kluter, P. Brisk, Y. Leblebici, and P. Ienne, "A design flow and evaluation framework for dpa-resistant instruction set extensions," in *Cryptographic Hardware and Embedded Systems-CHES 2009*. Springer, 2009, pp. 205–219.

[21] K. Tiri and I. Verbauwhede, "A digital design flow for secure integrated circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 7, pp. 1197–1208, 2006.

[22] P. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology—CRYPTO'96*. Springer, 1996, pp. 104–113.

[23] N. N. Mai-Khanh, T. Iizuka, A. Sasaki, M. Yamada, O. Morita, and K. Asada, "A near-field magnetic sensing system with high-spatial resolution and application for security of cryptographic lsis," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 4, pp. 840–848, 2015.

[24] J.-J. Quisquater and D. Samyde, "Electromagnetic analysis (ema): Measures and counter-measures for smart cards," *Smart Card Programming*

and Security, pp. 200–210, 2001.