UCLouvain

Ecole polytechnique de Louvain
ICTEAM Institute
UCLouvain Crypto Group

Noise Analysis for the Design of Side-Channel Secure Cryptographic Circuits

Kashif Nawaz

Thesis submitted in partial fulfillment
of the requirements for the degree of
Docteur en Sciences de l'Ingénieur

Composition du jury :
Prof. Denis Flandre (UCLouvain, Belgium) - Promoteur
Prof. François-Xavier Standaert (UCLouvain, Belgium) - co-Promoteur
Prof. David Bol (UCLouvain, Belgium) - President de Jury
Dr. Dina Kamel (UCLouvain, Belgium) - Secretary
Prof. Jean-Luc Danger (Telecom-ParisTech, France)
Prof. Fernando Silviera (Universidad de la Republica, Uruguay)
Prof. Itamar Levi (Bar Ilan University, Israel)

Louvain-la-Neuve, Belgium – September 2021

Dedicated to my parents, my dadi, my wife, my family, my brothers, my friends and my cats, muezza and bell

Acknowledgments

The journey of a PhD is tough, challenging and mostly lonely. However, during the course of my stay and work in Louvain-la-Neuve, I have been fortunate to be in the company of some very talented and good people who not only provided an academic and congenial atmosphere to work, but also contributed to in making the journey enjoyable and providing those moments which makes a graduate research life what it is. Although, I may not be able to name each and every person, but please believe that my gratitude is heartfelt and honest, no matter how big or small your contribution may have been in my journey.

At the very onset, I would like to thank my promoteurs, Prof. Denis Flandre and Prof. François-Xavier Standaert who have been pillars of strength and guidance in this journey. I am deeply honored to have been associated as your student and colleague and will forever cherish our discussions, meetings and tips you have provided me. Throughout the ups and downs of this rigorous academic life, you both have been the light which continued to inspire me and guide me. I truly feel blessed and very fortunate to have worked and been supervised by you. Thank you.

Thanks to my thesis committee members, Prof. David Bol and Dr. Dina Kamel who have always been a motivating and guiding me through the nuances of the PhD. Special thanks to Dina, who during the starting period of my PhD, helped me get familiar with the tools and workspaces and the itty-gritties of diving into a field which is so vast and interdisciplinary. Special thanks to Prof. David Bol, who has always been very accomodating for fruitful discussions and helpful tips. Thank you.

Thanks to the jury members for sharing their valuable time and providing enriching feedback in improving my thesis and being a part of my jury committee. Thank you.

Thanks to my office colleagues, who provided a very academic as well as a relaxed atmosphere in helping me continue my research. I am

fortunate that I had the opportunity to be associated and share office space with both the UCLouvain Crypto Group and the ECS group. I have learnt a lot through the discussions, bi-weekly seminars and lunches, presentations, and over the coffee table talks. Special thanks to Thomas, Santos, Davide, Itamar, Florentin, Francesco and Pierre. The latter half of my PhD I shared workspace with the ECS group and deservedly thank Léopold, Antonine, Jonathan, Roseline, Lamia, Sedki and Hajar for providing a fantastic work atmosphere. Thanks also to the ECS group members for putting up with my queries and bearing with me when sharing the server CPUs. Special thanks to Léopold, with whom I have shared many a discussions, related to topics beyond Electronics and Cryptography. Special thanks to Maxime, Thomas, Guerric, Ludovic and Charlotte who have always been very helpful and willing to help. Special thanks to Cecilia for always being a smiling and helpful colleague and who greatly assisted me during the teaching sessions. Special thanks to Valeriya for providing data related to initial device measurements and for all discussions. Thank you.

Special mention for my friend Wasil, who unfortunately left us all too soon and all too unexpectedly. You will be missed for all your help, kindness and cheerfulness which you brought everyday, both as a colleague and as a friend.

Thanks to my friends Raj and Ratan, who were there for the major part of my PhD and who have always been a source of motivation and help. From cooking Indian food together to discussing abstract research ideas, this will surely be missed. Thanks to all the friends in LLN for the cricket sessions, dinners and sports sessions. Special mention and thanks to Sahar for helping with the printing of the thesis and proof-reading the final version. Thank you.

Thanks to all the administrative and IT staff at the ICTEAM institute who have guided me through the administrative tasks and kept the servers, laptops and networks running making my journey smooth while continuously working in the background. Special thanks to Mme. Viviane Suavage and Mme. Brigtte Dupont for helping me throughout this journey.

Thanks to all the members of the ICTEAM institute who have been very respectful colleagues and directly or indirectly provided useful learning interactions and helped in any way as they could. Thank you.

Thanks to all the students for whom I conducted the teaching sessions. It has truly been an enriching experience and I hope the same for you too. Thank you.

Thanks to the Wallonia region for funding my grant and allowing me the opportunity to come and research in Belgium. I am truly indebted

to the people of Belgium for this. Thank you.

Thanks to my colleagues at the Technology Innovation Institute for extending their support during my PhD completion and supporting me to complete the journey. Thank you.

Thanks to my friends, Mushir and Ana, for always being a source of motivation during my PhD and for all the discussions and visits and trips.

Thanks to my family who have always been truly standing by my side patiently throughout the journey. I honestly, cannot thank you much. Thank you.

Last but not the least, thanks to my wife Shazia, for her constant support and motivation throughout the PhD. Thank you.

List of Figures

List of Tables

Abstract

Noise in digital circuits has always been minimized to achieve high signal integrity, robust operation and of course high performance. However, for cryptographic applications, increased noise can in fact be beneficial. It can be used effectively to reduce the (cryptographic) Signal-to-Noise (SNR) ratio and to make it harder for an adversary to extract useful information (*e.g.,* secret keys) from the side channel leakage data. A natural question concerns the extent to which intrinsic (internal) noise is required to improve security. In this manuscript, we explore this question and suggest ways to exploit the physical noise as a building block on which system level noise countermeasures can be further applied.

With this goal in mind and noting the fact that technology shrinking is simultaneously happening, we start by exploring the different transistor-level countermeasures available to cryptographic designers and the amount of signal which actually leaks from such implementations. Although this is a promising approach, nevertheless, it points us to the direction of exploration of noise for further lowering of the SNR and provides a framework for the concrete-work defined in this thesis. Namely, we try to answer the following question: is the noise generated from intrinsic device models a sufficient countermeasure on which higher (i.e. protocol or mathematical) levels countermeasures can be applied guaranteeing a base-level minimum noise? This noise should be difficult to filter out by an adversary, even with a sophisticated experimental set-up. More importantly, we try to answer the following questions: firstly, can it be `simulated` and `quantified`; secondly, could it allow digital designers to know well in advance the amount of noise guaranteed to achieve suitable design margins and finally, will it allow designers to be better prepared for post-layout and post-foundry simulations (and measurements) respectively.

In this work, we introduce this concrete simulation methodology to exploit the intrinsic physical noise (i.e., flicker and thermal noises) at the secure circuit level. We additionally demonstrate how the values

obtained from our methodology translate into relevant cryptographic metrics. Our simulations show that the calculated cryptographic noise values are in close agreement with the noise levels extracted from noisy distributions using transient noise analysis. We finally evaluate (with the proposed methodology) several meaningful parameters which affect the internal noise (and their security extent) such as transistors-sizing and voltage-supply changes.

We further explore the interdependence of such noise on the peripheral artefacts of a circuit-under-test, namely, wire inductances and capacitances (intrinsic) present due to coupling effects between the interconnects and the printed-circuit board (PCB), and the behavior of the intrinsic noise in such environments. Again, we stress on the importance of simulations to serve as a guide for allowing the designer (in the case, the digital designer as well as the PCB designer) to appreciate the fact that noise plays an important role in cryptographic designs and how it can be embedded into the design without compromising on functionality and performance issues (namely, signal integrity, power consumption, and delay etc.).

We finally conclude our thesis with our stated observations. Although, numerous works cite measurement results (but not limited to [51, 54, 84]) for cryptographic analysis, this is the first work, which uses simulation-based and relevant statistical analyses and post-processing based case studies to explore the impact of intrinsic device physical noise on digital cryptographic circuits. The goal of this dissertation work is to quantify the intrinsic noise inherently present in a system and explore its manifestations through which it may be analyzed in a cryptographic context. Additionally, we also believe that our work could aid cryptographic (digital) designers in analyzing the intrinsic security of design implementations early in the development cycle, especially during design-stage evaluations. For research and academia, our work can be extended to explore different avenues in which the simulation time can be reduced to cover larger cryptographic tests, taking into account the intrinsic physical noise coupled with reduced simulation costs (both in terms of time and complexity) as well as obtain a more holistic and consolidated view of the intrinsic security of such design implementations.

Contents

List of Publications

[1] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 299–320, 2017.

[2] Dina Kamel, Guerric de Streel, Santos Merino Del Pozo, Kashif Nawaz, François-Xavier Standaert, Denis Flandre, and David Bol. Towards securing low-power digital circuits with ultra-low-voltage Vdd randomizers. In *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, pages 233–248, 2016.

[3] Kashif Nawaz, Léopold Van Brandt, Itamar Levi, François-Xavier Standaert, and Denis Flandre. A security oriented transient-noise simulation methodology: Evaluation of intrinsic physical noise of cryptographic designs. *Integration*, 68:71–79, 2019.

[4] Kashif Nawaz, Leopold Van Brandt, François-Xavier Standaert, and Denis Flandre. Let's make it noisy: A simulation methodology for adding intrinsic physical noise to cryptographic designs.[2] In *14th Conference on Ph.D. Research in Microelectronics and Electronics, PRIME 2018, Prague, Czech Republic, July 2-5, 2018*, pages 61–64, 2018.

[5] Kashif Nawaz, Dinal Kamel, François-Xavier Standaert, and Denis Flandre. Scaling trends for dual-rail logic styles against side-channel attacks: A case-study. In *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris,*

[2] awarded the Best Student Paper Award at PRIME 2018

France, April 13-14, 2017, Revised Selected Papers, pages 19–33, 2017.

[6] Kashif Nawaz, Itamar Levi, François-Xavier Standaert, and Denis Flandre. A transient noise analysis of secured dual-rail based logic style. In *2018 New Generation of CAS, NGCAS 2018, Valletta, Malta, November 20-23, 2018*, pages 146–149, 2018.

Chapter 1

Introduction

1.1 Motivation

Physical security of cryptographic implementations is an escalating concern. With the growth of lightweight cryptographic implementations (e.g., IoT based applications, smart cards, etc.) and integration of more complex functionalities such as AI/Machine Learning (ML) into devices employing cryptography, secrecy of the cryptographic keys continues to be of paramount importance. One of the most common attacks hardware security designers face are Side-Channel Attacks (SCAs). Ever since the first reports of SCA by Kocher et al. [63], extensive work, both theoretically as well as experimentally, has been well covered in the past two decades of their appearance and sufficient countermeasures such as, but not limited to, masking (e.g., Threshold Implementations (TI) [91] or Domain-Oriented Masking (DOM) [75]), including shuffling [73] have appeared. Such countermeasures are mostly applied at the algorithmic/protocol level and have valid security proofs which back the security of the design under very sound mathematical propositions. However, design stage countermeasures (which are applied either at the gate, circuit or the transistor levels) may often not be evaluated for their side-channel resistance until the design is received back from the foundry post-fabrication and would indeed then be highly problematic (i.e., without analog testing). Post fabrication, the die is evaluated for side channel attacks and if the attack is easily successful, it (often) implies re-work and re-design by adding further countermeasures. This gap in evaluating the intrinsic side-channel resistance very (early) in the design phase is a critical problem faced by digital cryptographic designers. Cryptographic designers when evaluating the intrinsic side-channel resistance of implementations often face the dilemma in choosing a sound noise model. As

is often the cases reported in side-channel literature, external noise is added to a simulation setup, followed by an attack scenario and/or design of newer countermeasures. However, such selection of noise parameters grossly over (or under) estimates the security of the device as the designer has little (or no) knowledge about the already existing (inherent) noise in the implementation. If known, it allows the designer to estimate, with a certain degree of confidence, the amount of noise which needs to be added when adding countermeasures (either at the algorithmic, gate, circuit or transistor level). Thus, a proper evaluation and analysis of the (intrinsic) noise which is always occurring (and is an inherent property of the devices themselves) motivates the work in this thesis. Furthermore, since such evaluations *a priori* need to be done during early design stages of the development cycle, where simulations are mostly carried out, it is equally imperative that such noise analysis be carried out in the reference of a simulational context.

We have provided a broad overview and the necessary motivation for the fundamental questions this thesis endeavors to answer. We further divide the questions addressed in this thesis into the below areas and ask more relevant (and deeper) questions. These allow for the gradual development of the need of a simulation methodology for evaluating and analyzing the intrinsic noise of cryptographic implementations using a Computer-Aided Design (CAD) (or simulational) based approach. We now address these sub-questions (briefly) related to each individual area in their corresponding paragraphs.

Effect of Technology Scaling In a parallel environment, wherein semiconductor companies have rapidly invested in technology scaling, and gate lengths are shrinking every few months, a new concern has arisen, especially for cryptographic designers. Namely, how do such (cryptographic) implementations perform with technology scaling? Traditionally, Digital Cryptography and Nanoelectronics have maintained a generation lag between them. Nanoelectronic circuits, on account of their drive towards lower power and higher performance, adaptive control and ultra-voltage operations, manage to stay ahead of cryptographic implementations, which traditionally, rely more on the resilience of existing one-generation lagged process node. However, while this may be of interest to commercial entities implementing crypto designs on production scale, it is imperative, from a research perspective, to analyze the effect of such rapidly shrinkage of technology and its subsequent effect on such designs; and make way for accurate and reliable predictions to ease the transition to such lower- technologies. This leads to the following questions in the context of our thesis,

1. How do circuit-level countermeasures (more specifically, novel logic styles other than CMOS) and which have been around for almost over a decade now, would perform with the rapid shrinking of the size of the transistor (i.e., with technology scaling)?

2. Could design stage evaluations provide an overview of the intrinsic side-channel resistance of cryptographic implementations?

3. What are the trade-offs encountered, both in terms of design and simulational complexity (not to mention the costs attributed related to migration of newer technologies), when moving across technology nodes and expecting the same (or enhanced) level of security?

Noise Analysis and Simulation Methodologies Varying degrees of countermeasures exist in literature, that are able to withstand SCAs (both theoretically and practically) (details covered in section 2.7). Not limited to these externally applied (mathematical or algorithmic) countermeasures at the system-level, circuit-based countermeasures employ a larger area and additional components and usually incur a higher cost of operation, more complex management of the operations and provide limited security. Additionally, it may not be clear from the beginning of a design stage cycle, what is the target security level (of the design) to be achieved and this process is solely activated once the fabricated die/chip has been received back from the foundry, followed by evaluations.

Motivated by these statements, the question we now ask, *"Is there a methodology which is able to accurately model and estimate the noise, coming directly from the implementations? If so, is it possible to characterize the noise from a designer's perspective? Furthermore, can such noise be exploited for the purposes of side-channel evaluations to have a preliminary, firsthand pre-fabrication evaluation of the intrinsic security of the implementation under design?"* To enumerate,

1. Can the Intrinsic Physical Noise (IPN), which inherently exists in every transistor, be characterized, quantified and simulated to extract meaningful Figures-of-Merit (FoM), for e.g., in a side-channel evaluation application scenario?

2. From a simulational perspective, is it possible to characterize the security of a design implementation and make assumptions about the intrinsic side-channel resistance? More specifically, can such techniques lead to a first hand, preliminary stage evaluation of the cryptographic FoMs, building upon which enhanced countermeasures can be applied, before sending the design to fabrication?

3. Additionally, is it possible to increase the effects of such noise (including through simulations) to enhance the security of a given design, maintaining the performance levels of the implementation?

4. Through such simulations, is it possible to analyze the impact of, for instance, supply voltage variation or increase in design complexity on the cryptographic FoM in presence of IPN?

Extension of the Noise Analysis and Simulation Methodologies These observations and questions bring us to the final aspect of the thesis, namely, at a system level, is it possible to model the behavior of a side-channel based system, such as comprising an implementation, active probes (for voltage/current measurements), the role of Printed-Circuit Board artefacts, alongwith the intrinsic physical noise, and evaluate the FoMs? To enumerate,

1. How does the intrinsic physical noise scale up when moving from transistor based implementations to a system level implementation?

2. How does the intrinsic noise impact the security of implementations when considering additional effects, such as due to the measurement-based artefacts?

1.2 Thesis Outline

This dissertation is divided into seven (7) chapters. The current chapter, Chapter 1 provides the necessary introduction, motivation, the questions addressed as part of this thesis and presents a brief outline of the work.

Chapter 2 provides the reader with the necessary Nanoelectronics and Applied Cryptography background for an appreciable reading of the thesis.

Chapter 3 describes in detail the investigations into the scaling trends of CMOS and Dual-rail logic devices when moving to lower technology nodes.

Chapters 4 and 5 describe in complete (both mathematical and simulational) details the introduced methodology in this dissertation work.

Chapter 6 presents a detailed investigation into the design and simulation of a side-channel measurement set-up in the scope of our presented methodology.

We finally conclude our thesis with the conclusions and future research directions in Chapter 7. Finalizing our observations, the tools and

methodologies investigated and an indication of what could be an optimal continuing way to further the research from this work.

Chapter 2

Preliminaries and Background

This chapter details the preliminaries and lays the necessary (mathematical and introductory) background for this thesis work. Since this thesis combines aspects of semiconductor circuits (electronic engineering) and applied cryptography (computer science) it is very inter-disciplinary *and we make the reader familiar with aspects from both these diverse fields.*

2.1 CMOS Technology

The standard cell libraries and Process-Design Kit (PDK) employed during this thesis is a commercial semiconductor technology provided from a reputed industrial foundry, (but not limited to) STMicroelectronics [36]. With the rapid shrinkage of technology node (thanks to Moore's Law), speed and performance of chips is of foremost concern. Coupled with the fact that such high speeds and high performing chips are (mostly) power intensive, there has been a paradigm focus to shift to low power and low variability designs. Consequently, a single design optimization which achieves all, is not practically possible. So commercial chip-licensing vendors often employ a combination of techniques[1] to satisfy the broad spectrum of application requirements in their PDK or standard cell libraries. Commercial PDKs (for both 65nm and 28nm, which are utilized in the context of this thesis) often come in a variety of flavors, some of which favor lower operating voltages (for instance in the sub-threshold region), referred to as Low Voltage Transistors (LVT), which have a higher I_{ON} current, hence achieve higher performance at the expense of higher leakage and higher power consumption) compared to Standard

[1] such as adaptive body biasing to control PVT variations, providing IP cores for digital cells including analog IP cores etc.

Voltage Transistors (SVT) or Regular Voltage Transistors (RVT), which operate at standard operating voltages (above the threshold voltage of the transistor).

The Fully Depleted Semiconductor-On-Insulator (FDSOI) based MOSFET differs from the conventional Bulk based technology MOSFET (ref. Figure 2.1) in having the Buried Oxide (BOX) structure (i.e., an ultra thin layer, $\approx$ 25nm, of buried oxide, hence referred to as BOX embedded in the substrate. This BOX structure allows a complete separation of the substrate from the source and drain dopant region; thus effectively reducing the probability of the substrate leakage to very low, negligible values. Also the ultra-thin BOX allows for reduced short-channel effects such as Drain Induced Barrier Lowering (DIBL). Consequently, the effect of static leakage on the overall transistor design is reduced. Briefly, the Fully Depleted Semiconductor-On-Insulator (FDSOI) technology allows for the following advantages over conventional bulk-based Complementary Metal-Oxide Semiconductor (CMOS) technology

Fig. 2.1: Bulk based MOSFET vs FDSOI based MOSFET [2]

1. Better electrical control of the formed channel, i.e., the source-drain linkage which denotes the "link" between the source and the drain terminals, arising from the preceding word channel; the goal is to show that the electrical connection in the channel which is formed by linking the source and drain terminals, indeed is the link responsible for the movement of electrons denoted as $\rightarrow e \rightarrow$ in Figure 2.2.; we do not mention leakage (either transistor or cryptographic leakage) here.

2. Lower effect of DIBL, thanks to the BOX structure aforementioned, consequently leading to a

   Lower threshold voltage, V_{th} and a higher I_{on}/I_{off} ratio.

3. Decreased gate length, thanks to technology shrinking leading to faster device performance (due to the effect of the increased cut-off

frequency, f_T[13]).

Additionally, the BOX structure also allows for total dielectric isolation between the source region and the substrate and the drain region and the substrate. This provides the following advantages

1. Lower source-drain (SD) capacitances → this is of importance especially in faster switching digital circuits; lower capacitances lead to quicker switching and hence faster performance.

2. Thanks to the very thin film, the threshold voltage and channel current show less variations with temperature fluctuations.

3. Better power efficiency.

The FDSOI technology also allows for Adaptive Back-gate Body Biasing (ABB); i.e., it allows for control of the body bias (i.e., the 4th terminal of a conventional 4-port MOSFET device) as shown in Figure 2.2[2]. To enumerate,

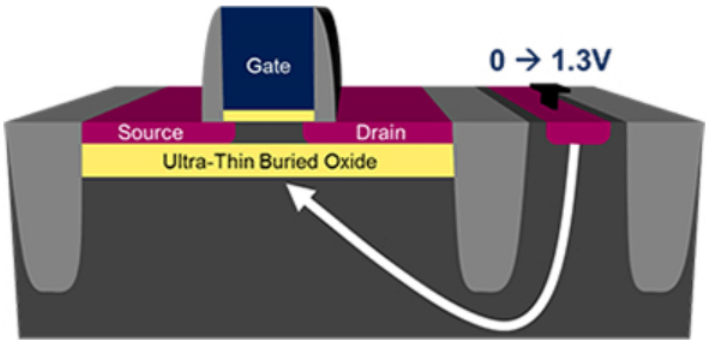


Fig. 2.2: ABB in 28nm FDSOI [2]

1. Adaptive control of the back-gate body bias to allow for both Forward Body/Back-gate Bias (FBB) and Reverse Body/Back-gate Bias (RBB) but not both on a given single device.

2. Increased speed, thanks to ABB, through V_{th} lowering in on mode, whereas V_{th} increases in stand-by mode.

3. Due to control of the body bias, the threshold voltages, V_{th}, of the device can be manipulated allowing for performance over wider range of voltages.

[2]Conventional Bulk-based CMOS also allow ABB; however, FDSOI allows for wider range of control the body bias

2.2 Power Consumption in CMOS-based Technologies

The power consumption of CMOS-based technologies (i.e., technologies which rely on the CMOS methodology of fabrication of the inherent MOSFET device) have power characteristics which make them characteristically different from other technologies (say, the BJT-based technologies).

From transistor theory, we are aware that the operation of a transistor in the inversion region follows two main modes of operation,

- the active (or the triode) region of operation where the current flowing through the transistor behaves linearly with respect to the applied voltage; i.e, the transistor behaves as a linear resistor and the resistance of the transistor is characterized as $\mathtt{R}_{ON}$.
- the saturation region, where the transistor current is "saturated" even when the applied drain-to-source voltage is increased. Current (constant) sources or voltage regulators need to operate in the saturation current to allow for a steady flow of the current (or voltage) across the device. Digital circuits operate between the cut-off and saturation regions whereas analog circuits in general exploit the saturation region to implement gain.

For (solely) illustrative purposes and to depict the various modes of operation aforementioned, we provide an example of a CMOS Inverter (the fundamental building block of digital integrated circuits (ICs)), as depicted in Figure 2.3 showing the the Voltage-Transfer Characteristics (VTC). However, we emphasize that the VTC is valid for the static region; hence not sufficiently clear. In dynamic operation, the power is dominated by the charge/discharge of the capacitors, and not the short-circuit current through the inverter as in the static case.

The characteristic CMOS power consumption is pre-dominantly `switching` dependent. For e.g., the power characteristic (or trace) which switches data from $0 \rightarrow 1$ or $1 \rightarrow 0$ will be inherently different from the switching (or rather the static switching) of $0 \rightarrow 0$ or $1 \rightarrow 1$.

The average power consumption[3] for a current *i(t)* flowing across a voltage *v(t)* to produce power *p(t)* can be represented as

$$p_{average} = \frac{1}{T}\int_0^T p(t)dt = \frac{V_{\mathrm{DD}}}{T}\int_0^T i(t)dt = V_{\mathrm{DD}} \times i_{average} \qquad (2.1)$$

[3]which consists of the switching dynamic component of the power

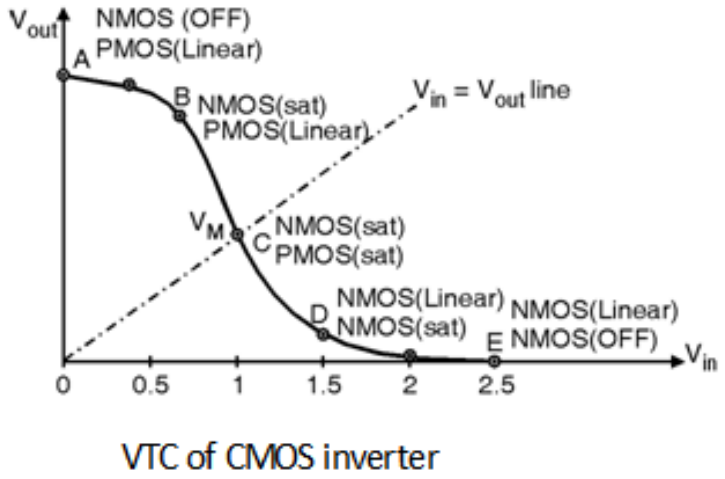


Fig. 2.3: Voltage Transfer Characteristics (VTC) of a CMOS Inverter [100]

The total power consumption due to a CMOS switching activity consists of both the dynamic and the static power contributions, i.e.,

$$P_{total} = P_{dynamic} + P_{static} \tag{2.2}$$

Equation 2.2 provides the power consumption due to all the transistors (in this case 2 for a CMOS Inverter) accounting for both the dynamic (transitions) or the static (no transitions) behavior. The sampled traces collected post-measurements (or post-simulations) are the sum of these two components. Post-processing, the traces need to be segregated (by application of some mathematical tools to segregate the dynamic and static regions) to further allow for analysis of the (individual) regions.

Dynamic Power Consumption Perhaps most relevant for side-channel analysis (and forming the bulk work of this thesis) is the dynamic power consumption of the current. This is the `switching` component of the current which changes due to a change of the inputs at one of input gate terminals (hence, data-dependent) and forms the majority of the total power P_{total}; however this claim has been shown to reverse in lower node technologies (i.e., sub-100nm) with an increase of the static leakage current only at very low V_{DD} (and hence the static leakage power, P_{static}) [80] or for circuits like SRAMs where a major part of the circuit is in hold, and only a small part is active (i.e., the read or write columns). The switching current component of the total current, $I_{switching}$ is responsible

for the charging and discharging of the output capacitance (i.e, for setting the output values)

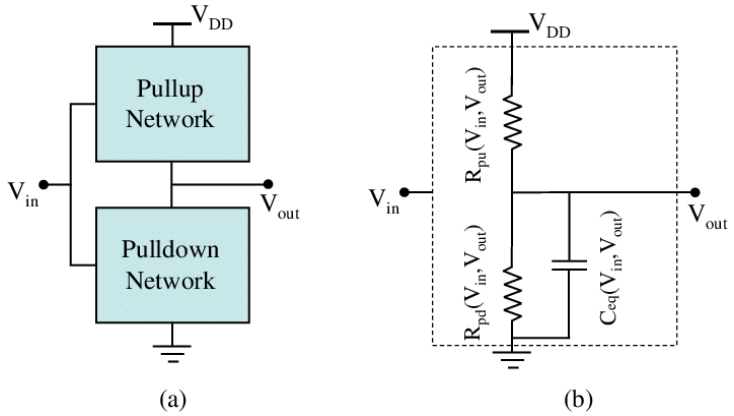


Fig. 2.4: Pull-up and Pull-down Network showing a Generic CMOS circuit architecture [60]

$$P_{dynamic} = P_{switching} + P_{sc} \tag{2.3}$$

where P_{sc} represents the short-circuit current due to the finite rise and fall times resulting in a direct path between the supply voltage and the GNDthrough the MOSFETs of the gate.

Since a CMOS-based digital logic can be represented as a combination of various inverter gates which ultimately form pull-down or pull-up networks (ref. Figure 2.4), the dynamic power component can be represented as the sum of all the switching components from the pull-up network and represented as follows:

Individually, the contribution of either of the networks can be represented as for a pull-up network (i.e, congruent to the PMOS devices of a CMOS gate (in general)),

$$\mathrm{F}(Input_1, Input_2, \cdots, Input_n) = 1 \tag{2.4}$$

Similarly, the contribution of the pull-down network related to the NMOS devices of the CMOS gate can be represented as

$$\mathrm{F}(Input_1, Input_2, \cdots, Input_n) = 0 \tag{2.5}$$

The overall output function, $(F)(Input_1, Input_2, \cdots, Input_n)$ is responsible for the total dynamic power consumption of the circuit.

For a system with N given inputs, the total switching power can be represented as,

$$P_{switching} = \frac{1}{T}\int_0^T V_{\mathrm{DD}} I^k_{switching}(t)dt = V^2_{\mathrm{DD}} f_{clk} C_{load} \alpha_F \tag{2.6}$$

where k represents the k_{th} input switching at time t, f_{clk} the frequency of operation of the circuit, C_{load} the total load capacitance, $pr^k_{0\to1}$ the corresponding switching probability of the gate/device in question and α_F the switching activity of the circuit represented as the ratio of the probability of the number of switching inputs to the total number of inputs, i.e.,

$$\alpha_F = \frac{\sum_{k=1}^{N} pr^k_{0\to1}}{N} \tag{2.7}$$

Short-Circuit Current The second component of the dynamic power consumption is provided by the *short-circuit* current, I_{sc}. From the VTC shown in Figure 2.3, when both the transistors are in the saturation region (i.e, they are ON and conducting) a short-circuit current flows across the power supply, V_{DD} to the GND ports of the device (or through the pull-up to the pull-down network in case of larger CMOS circuits). The transient behavior of this current, averaged over a period of time T_{sc} produces the short-circuit power component of the dynamic power consumption, represented as

$$P_{sc} = I_{average} V_{\mathrm{DD}} \tag{2.8}$$

where $\mathrm{I}_{average}$ represents the average value of the spike current flowing through the NMOS and PMOS on account of finite non-zero difference in the rise and fall times of an input switching signal.

Static Power Consumption Although, theoretically, the idea that CMOS devices do not consume static power has been around and was originally valid for higher technology nodes, but with the shrinking of gate lengths, the static leakage power has been shown to increase. Concerns arose at some point, when static leakage power had been shown to be dominant over the dynamic power consumption [59], but several circuit and device techniques avoided it to dominate. Further more, the advent to Moore 2.0 (Beyond Moore) has led to re-visit the transistor Metal-Oxide Semiconductor design and pave ways for further research, both from a semiconductor device level and an architectural level (e.g., 3D-stacking [14]).

Although 28nm FDSOI technology has claimed to feature lesser leakage power, thanks to the BOX design; nevertheless, further research

especially for cryptographic applications is required. Very briefly, we summarize here the contribution of the static leakage part to the total power consumed. The static leakage power can be represented as,

$$P_{leak} = I_{leak} V_{\mathrm{DD}} \tag{2.9}$$

where the current I_{leak}, referred to as the leakage current is the sum of the sub-threshold current. I_{sub} and the gate oxide leakage current, I_{ox}, represented as

$$I_{leak} = I_{sub} + I_{ox} \tag{2.10}$$

wherein $\mathrm{I}_{ox_{28}} << I_{ox_{65}}$, thanks to the high-k metal oxide gate and the sub-threshold current, I_{sub} for a device operating in the sub-threshold region (i.e., $V_{\mathrm{G}} < V_{\mathrm{th}}$) can be represented as

$$I_{sub} = K_1 W \exp\{-V_{th}/nV_\theta\}(1 - \exp\{-V/V_\theta\}) \tag{2.11}$$

where K_1 and n are experimental parameters, W is the gate width, and V_θ is the thermal voltage.

2.3 Noise Models

In this section, we briefly review the noise models used throughout the thesis. Additionally, other noise models (such as the Lorentzian noise i.e., Random Telegraph Noise (RTN) [127]) exist, but are outside the scope of this thesis. Noise simulations encountered in the thesis utilize the below mentioned models.

Thermal (and shot) noise can be modeled for simulations as a Gaussian noise current source connected in parallel to a noiseless element such as a resistor [130]. The thermal noise across the resistor can then be represented using the following equation,

$$i_{th}(t) = \sqrt{\frac{2kT}{R}}\zeta(t) \tag{2.12}$$

where k is the Boltzmann's constant, T the absolute temperature in °K and R the resistance. $\zeta(t)$ is a stationary white noise Gaussian process having a constant Power Spectral Density, PSD. For a MOSFET in strong inversion, the thermal noise can be represented as

$$I_{ntg}^2 = 4kT(\frac{2g_m}{3}) \tag{2.13}$$

in A^2/Hz where g_m is the small signal transconductance at the bias point.

Flicker noise is the dominant source of noise in MOS transistors at lower frequencies [33, 46, 45, 131, 132, 133]. It is called as $1/f$ noise as the spectrum varies as $1/f^{\alpha}$, where α, the flicker noise exponent has a value close to unity (α=1$\pm$0.2) The Power Spectral Density (PSD) of flicker noise in a MOSFET can be generalized as

$$S_i(f) = \frac{i_f^2}{\Delta f} = \frac{K_f}{C_{ox}^2 WL}\frac{g_m^2}{f} \tag{2.14}$$

where C_{ox} is the gate capacitance per unit area, W the channel width, L the channel length and f the frequency. The constant K_f is the flicker noise coefficient and is dependent on the process technology. Hence the PSD of flicker noise is varying as a function of the frequency. From (2.14), flicker noise cannot be directly used in a transient noise analysis as it is a non-stationary noise process. However, it can be synthesized as a RC circuit which sums up Lorentzian spectra and approximates it as a $1/f$ PSD [117, 118, 135] by

$$S(f) = \frac{2kT}{\pi C_m}\sum_{m=1}^{M}\frac{\varphi_m}{\varphi_m^2 + f^2} \propto \frac{1}{f} \tag{2.15}$$

where φ_m is the pole frequency of each Lorentzian spectrum and which can be represented as a white noise current source with a resistor R_m and capacitor C_m connected in parallel [64].

2.4 Transient Noise Simulations

In this section, we introduce the motivation for Transient Noise Analysis and present a didactic approach to introduce such simulations as offered by commercial vendor tools.

Motivation for Transient Noise Analysis Traditional analog noise simulations rely on the small-signal model of a circuit to simulate noise under `.AC` conditions for a given fixed reference bias point. While (such) frequency domain simulations (such as the `.NOISE AC` analysis) only provide the noise PSD, they offer no scope in the transient simulational analysis of complex digital gates, which often employ a large-signal model. While (only) transient analysis of such digital blocks provides validation of the functionality, their noise computations remain unknown in a transient setting. Characterization of such circuits in presence of IPN is only possible thanks to Transient Noise Analyses which incorporate transient noise current sources. Hence, the motivation for the need of

a transient noise simulation. In the context of this thesis, which is in a cryptographic context (and hence contains digital circuits), this provides a very useful tool indeed. It is noteworthy, that the principal attacks (detailed subsequently) on cryptographic hardware devices comprises of a class of attacks known as Side-Channel Analysis (SCA), which consists of measuring the power (or the current or the equivalent voltage) traces temporally; i.e., the value at each time sample is recorded and subsequent post-processing steps consists in applying advanced analysis to such (time-sampled) recorded instants.

Characteristics of Transient Noise Analysis Simulations Transient Noise simulations provide a simulational way of generating IPN from transistor sources and including the noise directly into the transient analysis simulations. Notably, the noise generated is added "intrinsically" to the simulation itself, (simulating a flicker and/or white thermal noise) rather than merely being added externally. This provides the following advantages:

- Since the noise is embedded directly into the circuit, it provides a better model for representing the various noise sources in the MOSFET.
- Theoretically, this noise is difficult to filter out as the transfer function is a complex system of multiple parameters.
- Although for theoretical assumptions, the noise can be assumed independent within each transient time sample (for instance, in static regions of operation, although not fully un-correlated due to bias dependence), the dynamic switching, on account of its faster switching time (in the orders of picoseconds), representing a time-varying correlated transfer function.

MOSFET Current Noise We now attempt to quantify the transient noise analysis using a more didactic approach using the commercial tool Eldo, provided by Mentor Graphics, as the primary simulator[4]. Each noise source can be represented as a instantaneous current source, $\mathrm{i}_{v_{DD}}(\mathrm{t})$ consisting of two components, the fixed (or the deterministic) current, $\mathrm{I}_{v_{DD}}(\mathrm{t})$ and the random (or stochastic or non-deterministic) varying component, $\mathrm{i}_n(\mathrm{t})$, which represents the noise generated *intrinsically* from MOSFET (or such based) devices.

[4]although we note, that Spectre, from Cadence Systems, provides a similar in-built transient noise analysis functionality, but for the purposes of this thesis, Eldo was the tool of choice throughout this work

Symbolically,

$$i_{v_{DD}}(t) = I_{v_{DD}}(t) + i_{v_{DD},n}(t) \tag{2.16}$$

The random component of the noise can be described as consisting of the following components: [128]

- a zero-mean $\mathrm{E}\{i_{v_{DD},n}(t)\} = 0$,
- a distribution of the amplitude, and
- a power spectral density, $\mathrm{S}\{i_{v_{DD},n}(f)\}$ (as described in section 2.3 by equation 2.14).

Generation of the Noise Current Sources We now, very didactically, introduce the characterization of such noise sources in commercially available simulators[5].

The current sources (which are representations of the physical noise sources) generate noise as the sum of a fixed number of sinusoids, N_t [18]

$$n(t) = \sum_{j=1}^{N_t} a_j(t) \cdot \sin(\omega_j t + \psi_j) \tag{2.17}$$

The stochastic property comes from the distribution of the phase ψ_j and the angular frequency ω_j, whereas a_j specifies the magnitude [18].

The value of $\mathtt{a}_j$ can be computed as follows:[18]

$$a_j = \sqrt{2 \cdot x_j} \tag{2.18}$$

where

$$x_j = \int_{f_{j-1}}^{f_j} S_j(f) \cdot df \tag{2.19}$$

where $\mathrm{S}_j(\mathrm{f})$ represents the PSD of the noise within the given frequency band f_{j-1} and f_j.

Temporally, as the simulation time proceeds, $\mathtt{a}_j$ is updated (depending on the bias of the MOSFET) for every time step. This leads to an integration of the noise in a non-stationary fashion as the model of the noise PSD is bias-dependent. The power of the signal generated is the same as the power computed from the noise PSD[6][18]. The limits f_{j-1} and f_j correspond to the `FMIN` and `FMAX` range of frequencies, detailed in chapter 4.

[5]Although we base our approach using the Eldo tool, we emphasize certain proprietary details are part of the vendor and hence adopt a very didactic approach, without specifying the complete details; we adopt a similar approach towards K_f computations also later in this thesis work; as such formulas and equations are very technology-vendor and process specific, and are covered under confidentiality agreements

[6]this is a very important observation as we later utilize this concept for validation

Computation of the Simulation Results A typical commercial-grade simulator employs a total of $N+1$ iterations to run a transient noise simulation of which the `first` run is a traditional, `noiseless` simulation and the remaining N are transient noises with the added randomized equivalent noise current sources [18]. Only the phase is changed between two noisy runs [18] (i.e., the ψ_j parameter introduced above) to achieve a random generation of the values; the frequency remains constant. At every time step, dt, the `RMS` noise value is computed as follows:

$$n(t) = \sqrt{\sum_{j=1}^{N} \left(V_j(t) - V_0(t)\right)^2} \tag{2.20}$$

where $\mathrm{V}_0(\mathrm{t})$ represents the noiseless transient response of the circuit-under simulation and $\mathrm{V}_j(\mathrm{t})$ the j_{th} transient response of the simulated circuit with the noise sources; and N the number of noise simulations performed [18].

2.5 Applied crypto background

The cryptographic aspect of the thesis provides the motivation to evaluate the impact of the aforementioned noise models in a cryptographic setting, i.e., we analyze the impact and role of the intrinsic noise on the security evaluations of cryptographic implementations, its quantitative assessment and its role lies in providing a building block for the application of additional countermeasures (both at the gate-level as well as system level).

In this section, we review the cryptographic tools necessary for the working of the thesis. We first introduce the XOR, PRESENT (4-bit Sbox) and Advanced Encryption Standard (8-bit Sub Bytes) designs which have been used throughout the thesis as the underlying implementations on which the noise analysis has been evaluated. We next introduce Side-Channel Analysis (SCA), the tools and methodologies relevant with respect to this thesis for SCA, and finally countermeasures which are used against SCA.

2.5.1 Cryptographic Design Implementations

2-bit XOR

A 2-bit Exclusive-OR (XOR) gate forms the most basic functional building block upon which classical cryptography relies. The 2-bit XOR performs the XOR, linear operation and is used as a building block

for block ciphers, permutation based crypto and stream ciphers. We represent a 2-bit XOR as Figure 2.5. Evaluating the performance or security of such a design does not make any practical sense; however, it serves as a useful block to evaluate the impact of intrinsic noise due to a basic gate, following which the noise in a larger implementation can be quantified.

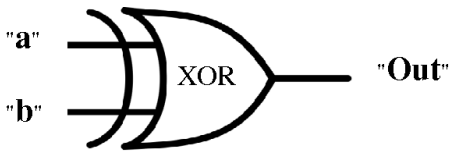


Fig. 10. Schematic of 2-input "XOR" gate.

TABLE II. TRUTH TABLE OF 2-INPUT "XOR" GATE

a	*b*	*Out*
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 2.5: A single 2-bit XOR [65]

Although, a single XOR by itself does not provide any security validation, a transient noise response (ref. Figure 2.6) on the application of a switching signal at the input provides the motivation to further explore this behavior. As depicted (ref. Figure 2.6), a simple 2-bit XOR gate is able to provide a variation of 50nA supply current[7] (at a given time sample) due to the presence of intrinsic physical noise. We specifically note here, there has been *no* other addition of any noise (additive or multiplicative or whatsoever) to the simulation of the transient noise response curve of the 2-bit XOR

PRESENT 4-bit Sbox

The PRESENT algorithm was one of the first standards targeting lightweight applications, specifically power-constrained and ultra-lightweight applications [34]. Designed by Bogdanov et al [15], the Sbox in PRESENT is the non-linear function. The PRESENT algorithm is often referred to as PRESENT-80, referring to the 80-bit key with a 64-bit plaintext. For the scope of this thesis, we focus on a single 4x4-bit Sbox (the PRESENT algorithm implements 16 such 4x4-bit Sboxes). The datapath design consists of a registered input and output ports implemented in Very High Speed Hardware Description Language (VHDL) using semi-custom

[7] for a given 28nm FDSOI process with C_{load}= 1fF

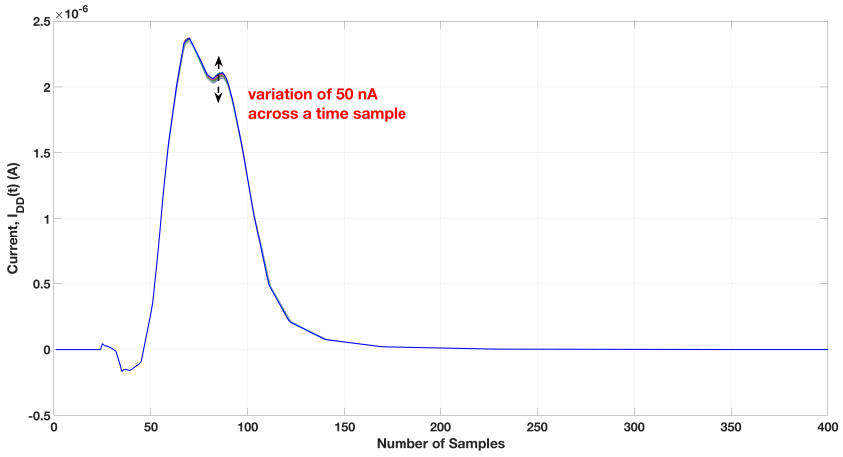


Fig. 2.6: Transient Noise response of a 2-bit noisy (including flicker and thermal noise sources) XOR for $V_{\mathrm{DD}} = 0.9\mathrm{V}$ at 298K across 100 traces

design flow and a transistor level design using the full custom design flow (targeting minimum sized transistors).

AES 8-bit Sbox

Since its standardization in 2001, the Advanced Encryption Standard (AES) [3] is the de-facto standard for encryption (and decryption) of symmetric security. AES operates in the block-cipher mode of operation (the details of which are not within the scope of the thesis and can be referred from [40]). AES operates as 3 variants: AES-128, which inputs 128-bit keys and provides 128-bit security level, AES-192, which inputs 192-bit keys and provides 192-bit security level, and AES-256, which inputs 256-bit keys and provides the highest symmetric security level, i.e., AES-256

All three designs implement the 8-bit Sub Bytes (Sbox), which is an invertible non-linear byte substitution, operating on each byte of the state separately. Although, traditional AES designs often implemented the Sbox as a Look-up Table (using memory RAM), the performance

degradation and under-utilization of resources, in case of an FPGA-based implementation, was high. Subsequently, a plethora of works has focused on the design and optimizations of the Sbox (and the AES overall, details of which can be found in [41]). For the scope of our thesis, we focus on the design using GF(2^m), where $m = 8$ [106, 77, 55]. Thanks to mapping the elements of the original field GF(2^8) to the composite field GF($(((2^2)^2)^2)$), the gate complexity and the power consumption can be greatly reduced. The adopted S-box consists of 3 stages: a transformation stage to map the elements to the GF($(((2^2)^2)^2)$) field, an inversion stage and an inverse transformation stage to map the elements back to GF(2^8), grouped with the affine transformation (ref. Figure 2.7).

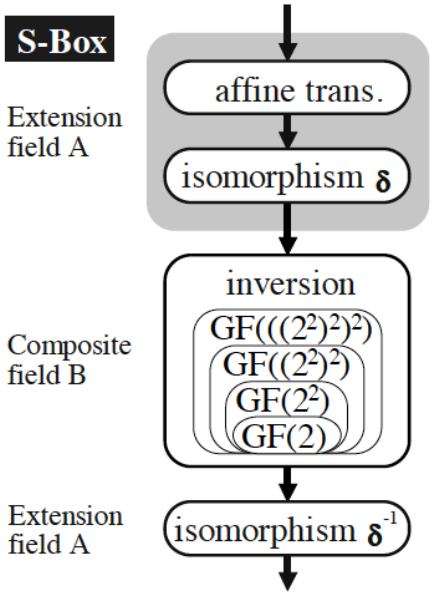


Fig. 2.7: Representation of the Transformations for the Implementation of the S-box [106]

The irreducible polynomial [106] utilized in the Rijndael S-box is

$$m(x) = x^8 + x^4 + x^3 + x + 1 \tag{2.21}$$

An outline of the S-box implementation [106] is depicted in Figure 2.8. The combinatorial design datapath (with clocked input and output registers) consists of both full custom and semi-custom designs. The CMOS implementation of the S-box is made only of 2-input AND/NAND and 2-input XOR/XNOR gates. The total number of transistors is 1,530, with a logic depth of 22 gates. On the other hand, the Dynamic

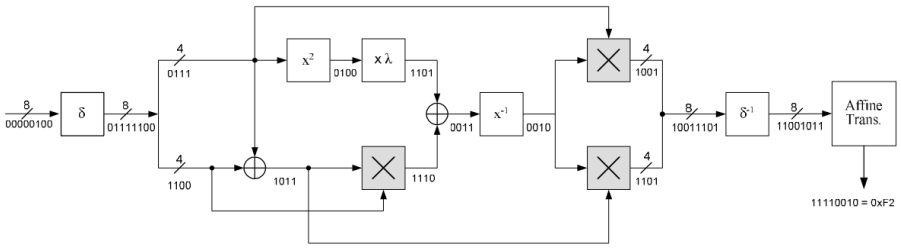


Fig. 2.8: Representation of the Transformations for the Implementation of the S-box [106]

Differential Swing Limited Logic (DDSLL) (DDSLL) S-box accounts for 1275 transistors, with a logic depth of 13 gates.

2.5.2 Side Channel Analysis - A Brief Context

The field of Side-Channel Analysis (SCA) has been existent for almost two decades and lays on solid foundational understandings (both mathematical and formal). However, from a hardware (or a more granular transistor-level) perspective, a number of questions still remains unanswered. Numerous works, exploiting multiple aspects of side-channel security are available in literature and subsequently analyzing every one of them (and which is not the goal of this thesis), is impractical; though to mention a relevant few as follows. In this section, we introduce the notion of side-channel attacks from a qualitative perspective, keeping in mind that for the scope of this thesis, side-channel is the background on which we base our findings; consequently, any practical such side-channel attacks, often encountered in literature, remain outside the scope of this work. We follow this by stating the relevant security evaluation metrics which have been used throughout this work. Although the metrics are by no means exhaustive, for the purposes of noise analysis, (which as such did not necessitate the need for "performing" side-channel attacks), metrics more relevant from an analysis perspective were employed. The goal here, being to quantitatively assess the impact of intrinsic noise originating from any cryptographic implementations.

Broadly speaking, side-channel attacks are defined as malicious adversarial attacks which exploit any residual side-channel emanating from a leaking implementation. We now explain each of these terms in more details:

- malicious: Although this term implies an attacker with "mischievous" intentions, for e.g., recovering the keys from a smart

card which unlocks doors to a secure server storage room, it can also be referred to for e.g., evaluators (in a cryptographic evaluation setting), whose job is to attack an implementation for the purpose of finding vulnerabilities, exposing them, and providing fixes so that the designer can fix back the issue.

- adversarial: This refers to an adversary (a.k.a. "the bad guy") who is capable of performing (or launching) a side-channel attack on the target device. It could be an individual hacker/hobbyist, a consortium, a group of hackers/evaluators or a nation-state sponsored actor. In short, any entity who is un-authorized to use the target implementation in ways not originally devised for with a "malicious" intent.

- side-channel (residual): The term side-channel refers to any aspect of the target device or implementation which inadvertently (or maliciously) leaks information which jeopardize the security of the device and which are outside the cryptographic model. Side-channels, as such, are visible when the implementation is operating (either dynamically or statically), and manifest as an additional accessory output; i.e., apart from the expected outputs which the design is expected to produce, it outputs additional information, which if correctly exploited can be utilized by a malicious adversary, could lead to details about the design and retrieval of useful information for e.g., cryptographic keys. Side-channels, from digital implementations as such, include power consumption [63][73], electromagnetic emanations[5] [37][57], thermal signatures [40][48][76], optical[35][116], timing information[22][62] or acoustic sounds[9][125].

- leaking implementation: This refers to any implementation which has side channel information "leaked from any of the aforementioned channels; depending upon the side-channel giving away, the information leaked could be current (or voltage values) (for power side-channels), EM radiation, optical signals or CPU usage sounds (for acoustic). For the scope of this thesis, we will focus only on power side-channels.

2.5.3 Power Attacks in Side Channel Evaluation

Power Analysis Attacks (or PAAs) form one of the most common modes of attacks in the side channel community targeting cryptosystems by exploiting the side-channel leakage power. By leakage, we *do not* refer

to the static leakage power of a MOSFET device; rather in applied cryptography, and more specifically, side channel analysis, `leakage` refers to physical entity which emanates from the side channel (as defined above). For a leakage to be relevant and useful to an adversary, two criteria need to be sufficed [32]

1. a leakage model, namely, a model of the leaking implementation which converts the leaked power (or current or voltage values) into their corresponding data-dependent cryptographically relevant values, for e.g., Hamming Weight (HW) and Hamming Distance (HD) based leakage models, normally used for microcontrollers and Field Programmable Gate Array based designs (including Application Specific Integrated Circuit designs).

2. Secondly, a relevant distinguisher, i.e., a statistical tool which is able to distinguish and highlight the differences (or rather the points of differences) between the classes (or sets) of data obtained as a result of side channel attack.

Although numerous works in literature, [32] [31] [69] [114], focus on both the aspects, the roles of an evaluator and a designer are often blurred. For an evaluator, whose goal is to certify the leakage of a chip, will nevertheless need to model the leakage from the chip correctly, and then employ the best possible distinguisher. For a designer, on the other hand, the design strategy would be to obfuscate the design as such that it is not easily expressed by a leakage model. Although both designers and evaluators work as adversaries, their end goal remains the same, i.e., to enhance the security of the implementation against side-channel adversaries. Throughout the scope of the work, we switch these roles interchangeably.

Power Analysis The leakage from a side-channel attack can be expressed as the summation of the individual components which contribute to the total power. Thanks to the data-dependency behavior of CMOS based devices, we attribute each power value to a sample, k, such that

$$P_{tot}[k] = P_{data}[k] + P_{el}[k] + P_{sw}[k] = P_{data}[k] + P_{noise}[k] \tag{2.22}$$

, where P_{tot} refers to the total power consumption of the device/chip [73], P_{data} refers to the component of the total power consumption due to the switching of the input (or transient) signal, P_{el} refers to the electronic noise inherent in the measurement set-up, and P_{sw} refers to the switching noise which are variations in the power traces that are caused by components (or cells) not relevant from a side-channel attack;

for instance, the switching power caused due to the buffers or I/Os which may not be connected to the V_{DD} as the cryptographic implementation [73]. Conventionally, the latter two components are aggregated (or summed together) and referred to as the noise in the measurement (or observational) set-up.

The data-dependent portion of Equation 2.22 [[73] contains the information which the adversary wants to extract. The other portion, referred to as the Noise Power or P_{noise} is attributed to any such artefacts which are accessory to the device under attack; commonly, these include measurement artefacts, noise due to electronic switching (of the charge carriers), switching noise, algorithmic noise[8] (not shown above) and even intrinsic physical noise. We briefly describe them below. For a more exhaustive read, we refer the reader to [73].

Electronic Noise The electronic noise, P_{el} from Equation 2.22 is mostly normal distributed (including for cryptographic devices) [73]. Fluctuations in multiple repeated observations (either via simulations or direct measurements) of data (i.e., voltage or current values) are attributed to this electronic noise. From an electronics perspective, this noise (mostly) comprises of flicker and white noise, ref section 2.3.

Data-dependency The data dependency, P_{data} from Equation 2.22 refers to the linkage between the power consumption of the cryptographic device and the input data it processes. In general for cryptographic devices, the input data to be processed is modeled as a uniform distribution.

$$\begin{aligned} P_{tot}[k] \circ P_{intrinsic}[k] = ((P_{data}[k] \circ P_{intrinsic}[k]) + (P_{el}[k] \circ P_{intrinsic}[k]) + \\ (P_{sw}[k] \circ P_{intrinsic}[k]) + \\ (P_{algorithmic}[k] \circ P_{intrinsic}[k])) \end{aligned} \tag{2.23}$$

$$\begin{aligned} P_{noise_{total}}[k] = ((P_{el}[k], P_{intrinsic}[k]) + (P_{sw}[k], P_{intrinsic}[k]) + \\ (P_{algorithmic}[k], P_{intrinsic}[k])) \circ P_{intrinsic}[k] \end{aligned} \tag{2.24}$$

Equations 2.23 and 2.24 provide the first introductions of *intrinsic physical noise* (IPN) as an additional source of noise. For the purposes of this thesis, we *only* consider the electronic noise (i.e., the Intrinsic Physical noise (IPN) coming directly from the MOSFETs). The IPN has the following characteristics

[8] noise from other components

- IPN is composite in nature. Compared to additive white Gaussian Noise (AWGN), this noise "embeds" itself with each component of the power (or current or voltage), both of the data-dependent power and the separate noise components.

- Due to its intrinsic embedded nature, filtering out the noise (without filtering the signal also) becomes a challenging task for an adversary. Any decrease in the noise (due to a filtering setup say, employed by an adversary) will filter out the information carrying components as well.

- The common assumption of assuming the probability density functions of noise components being Gaussian in nature (which relaxes the mean and variance assumptions) does not hold true when IPN is considered. Rather, as we shall see later, this leads to rather behaviors as Gaussian mixtures.

An Explanation of the Composite (or the Multiplicative) nature of the IPN Although IPN has been the subject of various research topics literature, individually, for instance as in flicker and white noise, its contextualization in a cryptographic scenario remains unexplored to the best of our knowledge and is the first of its work, as introduced in this thesis. Understandably, and noting the fact that, real world measurements of such noises are limited to basic devices (such as a single MOSFET, as utilized later in this work), produces a concrete problem in modeling the characteristics of such noise for cryptographic applications. However, on the course of this thesis, an understanding has been developed with respect to the nature of such noise and to formally quantify it.

Starting from Equation 2.23 and 2.24 [136], we note,

$$\begin{aligned} P_{out}(t) &= f(P_{data(t)}, intrinsic_{noise}(t)) \\ &= f(P_{data}(t) \circ P_{intrinsic}(t)) \\ = f(P_{dynamic}(t) \circ P_{intrinsic}(t)) &+ f(P_{static}(t) \circ P_{intrinsic}(t)) \end{aligned} \tag{2.25}$$

In the above equation, $\mathrm{P}_{out}(\mathrm{t})$ is the total value of the power, $\mathrm{P}_{data}(\mathrm{t})$ represents the data-dependent dynamic and static switching components and $\mathrm{P}_{intrinsic}(\mathrm{t})$ is the component which represents the power due to the additional non-input data dependent[9]. A rigorous mathematical

[9]although we stress here, that by independence, we do not concretely say that it is non-correlated also

treatment of the non-linearities of the power series terms are outside the scope of this thesis; however, we do provide an intuition of the apparent behavior. Rather, our goal here is to make the reader appreciate the effects of the non-linearities of the IPN and its correlation with the input switching data of a cryptographic circuit.

Using Equation 2.25 as a template and canonically replacing the power components by the variances of the noise power and expanding as above, we arrive at the below,

$$\begin{aligned} \sigma^2_{total} &= f(\sigma^2_{data} \circ \sigma^2_{\mathsf{IPN}}) \\ &= \underbrace{f(\sigma^2_{dynamic}(t) \circ \sigma^2_{intrinsic}(t))}_{\text{dynamic data-dependent}} + \\ &\qquad \underbrace{f(\sigma^2_{static}(t) \circ \sigma^2_{intrinsic}(t))}_{\text{static data-dependent}} \\ &= \left(\alpha_F(\sigma^2_{dynamic} \circ \sigma^2_{intrinsic}(t))\right) + \\ &\underbrace{\left(K'_f \sum_1^{N_T} \int_0^{fmax} \frac{g_m(f)^2}{f} df\right)}_{\text{represents the multiplicative component of the IPN}} \end{aligned} \tag{2.26}$$

, where K'_f represents the $\frac{K_f}{C^2_{ox}WL}$, from Equation 2.14 and the function $\frac{g_m(f)^2}{f}$ represents the non-linear dependence of g_m a a function of the varying frequency. The $\circ$ operation in the non-linear dependent part of Equation 4.29 represents the *multiplicative* part of the IPN contributing to the total noise variance (power), and the data-dependent part represents the *additive* part. The $\sum_1^{N_T}$ sums up the contributions of the various transistors, for the given activity factor, α_F, which form a complex non-linear transfer function. Since the individual calculations of the varying g_m's for any given transition is a complex and rigorous task, we sum up the total contributions of all the transitions and average to provide a consistent mean value (refer Equation 4.39).

The function, $f(\sigma_{data})$ represents the noise power variance due to the data-dependent (power) switching component, i.e., $f(P_{data})$ referenced in Equation 4.27. Notably, $f(\sigma_{data})$, like $f(P_{data})$ is a function of the number of switching transistors (and activity factor, ref. Equation 4.28). The non-linearity in the case of dynamic component arises from the complex interaction of the time varying load capacitance, $C_{load}(t)$ with the IPN, which is represented by the $\circ$ operation and is a function of the number of transistors, N_T and the activity (switching) factor, α_F (ref.Equation 4.28).

We compute the total variances, using both the ensemble averaging method and the total noise power computations using the PSD and attempt to reconcile to Equation 4.29.

2.5.4 Univariate Gaussian Distribution

Side-channel analysis relies on statistical tools for successfully performing an attack. Most of the classical side-channel based attacks such as the Simple Power Analysis (SPA) [62], Differential Power Analysis (DPA) [63], Template Attacks (TA) [24], Correlation Power Analysis (CPA) [20] use classical statistical univariate tools to employ such "distinguishers" which are able to distinguish between two (or any such segregated groups of data allowing for maximization of a resultant value, i.e., the *most probable or most likely or maximal chance*, which indicates the sample in time under interest where the switching of the current has taken place due to addition of a key etc. A statistical distribution showing the probabilities (or the number of occurrences) for a given set of current (or voltage values) indicates the mean and variance of a distribution [21]; i.e., the *expected* value of the current (or the voltage) in terms of a probability; similarly the variance of the distribution indicates the extent (or the *spread*) of the measured (or simulated) parameter. We illustrate this with a simple example.

Assume a measured (or simulated) current (or voltage) value. [10]. At any given instant of time sample, t, the total leakage can be represented as follows,

Let $\overrightarrow{L}$ denote a physical leakage (e.g. current trace) from a crypto implementation:

$$\overrightarrow{L(t)} = (\sum_{j=0}^{N_{inputs}} f_j(\overrightarrow{X}_{\text{input}}(\text{t}), \sigma_{intrinsic}(t))) + \overrightarrow{N}_{\text{physical}}(\mu, \sigma^2) \tag{2.27}$$

$$\sigma^2_{intrinsic} = \sigma^2_{flicker} + \sigma^2_{thermal} \tag{2.28}$$

where $\sigma_{\text{intrinsic}}$ is the variance due to the *intrinsic* noise produced by the MOSFET devices, $f(\overrightarrow{X}_{\text{input}}, \sigma_{intrinsic}))$ represents the cryptographic function being implemented operated on the X_{input}; and under the assumption of an additive Gaussian noise due to the measurement setup, $\overrightarrow{N}_{\text{physical}}(\mu, \sigma^2)$.

[10] For the sake of this thesis, henceforth, we will be referring to such current values as leakage; when referring to static leakage in case of electronic devices theory, we make it explicitly clear

A Gaussian distribution of the leakage parameter $\overrightarrow{L(t)}$ (simulated for random values) is shown in Figure 2.9. Clearly, the dependence of the input is observable on the leakage.

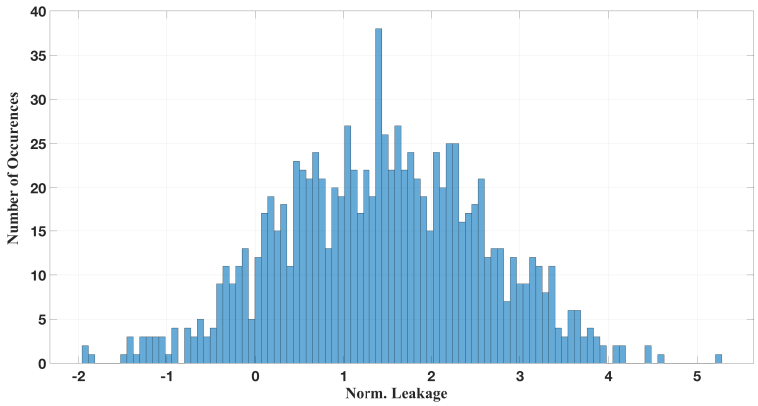


Fig. 2.9: Gaussian Distribution Plot for random simulated values with μ=Hamming Weights (from 0 to 8) and σ=0.8 for 1000 samples

Mathematically, a Normal (or Gaussian PDF) can be represented as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \tag{2.29}$$

and the curve in Figure 2.9 describes the probabilities associated with different values of the simulated parameter, for e.g., x, in this case. These first (such as the mean) (or second order moments e.g., the covariance) can be used to compare and differentiate distributions. Even when not considering an attack based scenario (such as within the scope of this thesis), these first, second (or even higher order) moments provide useful (in many cases, visual) representation to clearly distinguish between sets (or classes) of data upon which incremental changes have been made. Such a useful illustration is re-visited in Chapter 6 of this thesis.

2.5.5 Evaluation Metrics in Side channel Analysis

Evaluating the side-channel resistance of a leaking cryptographic implementation necessitates using a number of metrics in tandem with each other; often a single metric, by itself can not account for (or detail) the various leakages found in a system. For e.g., while the SNR metric (discussed subsequently) is a first-hand univariate metric, it does not provide information on higher order leakages (which as such may employ

higher order moments beyond the mean and variance). Additionally, being univariate in nature (i.e., the focus is on a *single* sample in time), inter-(or intra-) correlations between the samples (in time) i.e., multivariate, are often missed by such metrics. So metrics employing more complex statistical tools like covariances, (skewness and/or kurtosis) are applied. Furthermore, these metrics often rely on an assumption of the underlying leakage model (mostly, a Gaussian distribution); however, often in real-life scenarios where the device may not be available to the adversary before-hand (in order for him to create a template profile), information-theoretic (IT) based metrics are better as they do not rely on an underlying assumption of the leakage model of the device. We now briefly provide the reader an overview of the various metrics which have been employed in the scope of this thesis.

Signal-to-Noise Ratio (SNR) A quick state-of-art metric for quantifying the cryptographic leakage from a side channel (a leaky implementation) arises from the classical univariate metric, the Signal-to-Noise Ratio (SNR). In this paper, we use Mangard's Signal-to-Noise Ratio (SNR) defined in [72] as:

$$\mathrm{SNR} = \frac{\hat{\mathsf{var}}_x(\hat{\mathsf{E}}_i(L_x^i))}{\hat{\mathsf{E}}_x(\hat{\mathsf{var}}_i(L_x^i))}, \tag{2.30}$$

where $\hat{\mathsf{E}}$ (resp. $\hat{\mathsf{var}}$) denotes the sample mean (resp. variance) operator and L the leakage. In our following simulations, this SNR will be computed for noise-based traces of the current consumption of the digital gates at the supply rail as a function of time, denoted as $I_{DD}\,(t^*)$ and would include the noise coming from physical *intrinsic* MOSFET noise sources. Using eqn (2.30), the signal is the "useful" part that is obtained by the adversary as a measure of the information leakage. The lower the signal value, the lower is the "perceived" side-channel leakage.

Mutual Information (MI) Evaluating the side-channel resistance of a leaking implementation is a challenging task. Without a concrete model modeling the power leakage, it amounts to collection of a huge number of traces and employing multi statistical distinguishers. Additionally, once the device has been changed, this amount to repeating the process. Since different devices leak differently [79] [107] [113], having a uniform model is impractical. However, a concrete evaluation methodology which is able to evaluate the information leakage irrespective of the inherent device power leakage model, was proposed by Standaert et al. as the Mutual Information [114] based on Information Theory [107].

Let L_q be a random vector representing the leakage functions obtained after q queries (in this case, noisy simulated runs). Let l_q be a realization

For our work, where we consider the leakage function to be Gaussian distributed[11], the SNR and MI metrics are equivalent [30]. We use the MI as a better metric to visualize the security level as a function of the (*intrinsic*) noise variance.

$$\mathrm{MI}(X;L) = \mathrm{H[X]} - \sum_{\mathrm{x}\in\mathrm{X}} \Pr[x] \sum_{\mathrm{l}\in\mathrm{L}} \Pr_{\mathrm{simu}}[l|x].\log_2 \Pr_{\mathrm{simu}}[x|l] \tag{2.31}$$

where, H denotes the entropy, Pr[x] denotes the probability of the input variable x, $Pr_{simu}[l|x]$ is the conditional probability of the leaked variable for a given (simulated) input, and $\Pr_{\mathrm{simu}}[x|l]$ the conditional probability of the input given the (simulated) leakage.

The relation between the SNR and MI metrics can be stated as follows

$$\mathrm{MI(X;L)} = -\frac{1}{2}\cdot\log_2\left(1-\left(\frac{1}{\sqrt{1+\frac{1}{\mathrm{SNR}}}}\right)\right)^2 \tag{2.32}$$

Perceived Information (PI) The Perceived Information (PI), defined in [105] and formalized in [114] provides an approximation to the "average case" often encountered in a real-world attack (or evaluation) scenario. Since the true MI of a device cannot actually be determined, the PI provides the best case approximation of the leakage (under a reasonable Gaussian hypothesis) observed using the model employed by the adversary (the assumption being that the actual "true" model can never be known by an adversary; which corresponds to the MI).

$$\widehat{\mathrm{PI}}(\mathrm{X;L}) = \mathrm{H[X]} - \sum_{\mathrm{x}\in\mathrm{X}} \Pr[x] \sum_{\mathrm{l}\in\mathrm{L}} \Pr_{\mathrm{simu}}[l|x].\log_2 \Pr_{\mathrm{model}}[x|l] \tag{2.33}$$

Therefore,

$$\widehat{\mathrm{PI}}(\mathrm{X;L}) \leq \mathrm{MI(X;L)} \tag{2.34}$$

Principal Component Analysis Principal Component Analysis (PCA) [47] is a pre-processing technique employed to filter/compress a data-set with noise reduction and dimensionality reduction by maximizing along the variance. It works by transforming a large data-set into smaller ones with the principal (or the first one) containing the maximum information,

[11] strictly speaking, this holds for higher voltages; at close to threshold voltages or sub-threshold voltages, the distribution could indeed be skewed due to the relatively long delays encountered typically at lower or (sub)-threshold voltages; nevertheless, for the purposes of this thesis, we stick with the above assumptions

and the remaining (i.e., the secondary) can be disregarded as noise[12]. In terms of side-channel literature, it works by projecting the traces (i.e., the collected traces post measurements/simulations) into a new-data set using a basis comprised of the eigen vectors of a covariance matrix of the data-set. PCA allows to reduce the amount of noise while retaining only those components which hold the maximum information (for instance, in a side-channel leaking scenario, the principal component is the one along which contains the key leakage, while the rest are ignored as noise). The following steps are needed to be performed in order to make such an analysis[112].

- computation of the mean from the data set followed by subtraction from each of the dimension n for each trace T_i ; i.e., the mean vector M_n is computed along the n dimensions of the trace T_i

$$\begin{aligned} M_n &= \frac{\sum_{j=1}^{n} T_{i,n}}{n} \\ T_{i,n} &= T_{i.n} - M_n \end{aligned} \tag{2.35}$$

- construction of a covariance matrix, Σ.

$$\Sigma^{n*n} = \Big(c_{i,j}, c_{i,j} = Cov(Dim_i, Dim_j)\Big) \tag{2.36}$$

  where Dim_k is the kth dimension.

- computation of the eigenvectors and the eigenvalues of the covariance matrix,

$$\Sigma = U * \Delta * U^{-1} \tag{2.37}$$

  where $Delta$ is the diagonal eigenvalue matrix and $\mathcal{U}$ the eigenvector of Σ.

The component corresponding to direction of the maximum variance (i.e., the eigenvector corresponding to the largest eigenvalue), is the first principal component. Choosing n eigenvectors produces n principal components. Similarly, k such components can be retained, each as a vector in the columns of a matrix, called as a feature vector.

For SCA applications, the PCA (steps mentioned above) can be employed as follows:

PCA transformation This consists of multiplying the feature vector produced above with the original data $\mathcal{X}$

$$U_d = U_r * U_r^T L_d = T_i * U_d \tag{2.38}$$

[12]not to be confused with analog noise or cryptographic noise or IPN whatsoever

Noise Reduction Choosing only along the principal components without reducing the dimensionality can also be achieved to retain only the most useful information. Since no components are discarded, the original data is returned. This consists of transposing the feature vector $\mathcal{U}$, followed by multiplication with the transposed mean-adjusted data $\mathfrak{X}$. To return the original vector, this is multiplied by the feature vector again. A final transpose leads to the matrix $\mathfrak{Z}$, followed by a final addition of the mean [44].

$$Y_T = X * U \tag{2.39}$$

$$Z = U * (X * U)^T = U * U^T * X^T = X^T \tag{2.40}$$

$$\hat{X} = Z^T = (X^T)^T = X \tag{2.41}$$

TVLA TVLA or Test-Vector Leakage Assessment is a methodology which allows for the leakage detection from a given set of traces; in the context of this thesis we make use of the "fixed" vs "random" comparison of a given set of traces against a known set of traces, N_1. Although the technique is used in cryptographic evaluations to determine, qualitatively, the security of both unprotected and protected (masking) scheme, we employ this tool, as a first hand, to validate the usefulness of our methodology across the different use-cases explained in the subsequent section. By utilizing the first and second order statistical moments using a standard Welch's t-test rule, the traces under test are compared with a fixed set of traces, and the results tabulated/plotted as the maximum value over time which indicates the presence of leakage in an implementation. For our case study, since we explore the effects of the PCB artefacts on un-protected implementations, we naturally, observe the maximum value (over time) values above 4.5 However, we show here the change in the maximum (over time) value (which indicates the change in the ease of the leakage being detected) is corresponding to the degree of the artefacts being introduced in our model. Mathematically, for given sets of traces, N_1 and N_2, with the corresponding means μ_1 and μ_2, and their corresponding variances, σ_1 and σ_2, respectively, the maximum value t, for a given sample, i, is defined as

$$\mathrm{t}^{\mathrm{i}} = \frac{\mu_1^i - \mu_2^i}{\sqrt{\frac{(\sigma_1^i)^2}{N_1} + \frac{(\sigma_2^i)^2}{N_2}}} \tag{2.42}$$

2.6 Challenges for Side-Channel Security in terms of Noise Simulation Analysis

Side-Channel Analysis (and/or attacks) rely on the measurement of traces from a device-under-test (running a cryptographic algorithm) followed by post-processing of the same to recover the secret key. These traces are often "noisy", which includes in addition to the noise introduced from PCB artefacts, measurement artefacts etc., noise or errors introduced due to measurement bias. Often these types of errors, are simply removed by averaging (i.e., averaging over multiple traces, hence the need to collect more traces). Furthermore, artefacts-based noise (injected die to measurement artefacts etc.) could be removed by employing a low-pass filtering setup.

However, what becomes challenging for a SCA adversary is to remove the embedded noise (or the IPN) from the collated traces as this noise is embedded within the devices and forms part of the device's behavior. Consequently, the adversary is unable to "see" this noise and hence not able to filter it out, without significant loss of information (i.e., the useful part which contains details of the secret to be recovered). For cryptographic designers, the goal would be to maximize the effects of this noise (without compromising performance) so as to achieve the lowest SNR possible (i.e., the minimum detectable useful signal).

However, analyzing the impact of such noise is not very direct and is a complex problem. We enumerate the following challenges often encountered during transient noise simulations in the context of side-channel security,

1. The designer has to ensure that IPN is really a part of the noise which is coming from the devices themselves (i.e., the MOSFET in this case). Consequently, transient noise simulations have to ensure that the simulated noise current sources are intrinsic to the design and target (and correlate with) the MOSFET.

2. The levels of the noise should not affect the working performances of the devices as a whole; i.e., the noise should not be detrimental to the behavior (or the performance) of the digital circuit. Transient noise simulations need to ensure the sanity of the noise levels introduced so as to not perturb the normal operation of the device.

3. Analysis of such a noise should ensure that noise values can indeed be computed and simulated beforehand to ensure their behavior before the chip/designs goes into fabrication. Subsequently, transient noise simulations should offer a forecast of the noise expected

from a design post-fabrication. In the context of side-channel security, this is imperative as this allows the designers to evaluate the intrinsic security of a physical implementation (during design stage) before adding external (higher-level) countermeasures (either during design time or post-fabrication in case, of system-level countermeasures).

4. The noise should be scalable with the design; i.e., as the design complexity grows, so should the corresponding noise power increase so as to maintain similar levels of SNR (without compromising the performance). Transient noise simulations should scale correspondingly with the increase in design complexity.

5. Transient noise simulations should emulate the real-world working of a chip (design) where correlations between adjacent (time) samples are common, for instance, thanks to jitter introduced by a noisy clock. Although it is still possible for a worst-case adversary to filter out the noise (either though sophisticated filtering setups or by simply averaging over a large set of traces or by alignment techniques), the overall complexity of the attack, nevertheless grows with such temporal correlations; thus increasing the complexity of such attacks in practice.

6. The simulations for such analysis should allow for a mechanism to alleviate the noise values (although hypothetical and mostly restricted to simulations only) in order to understand the maximum limits of noise that could be introduced into a system without compromising its functionality and without performance overheads.

2.7 Countermeasures

A number of countermeasures exist in literature; a full survey or details of which are not within the scope of this thesis; [73] provides a decent introduction. Countermeasures can be applied at various design levels of hierarchy, which can be tabulated as Table 2.1.

From equation 2.30, it is clear that to decrease the information leakage, the numerator must decrease (i.e., the signal must be decreased) or the denominator must be increased (i.e., the noise of the implementation should increase). Algorithmic or Gate-level countermeasures principally employ the latter technique (i.e., by adding random bits, shuffling or adding algorithmic noise). Circuit level countermeasures, which will be discussed next, employ circuit level techniques such as supply voltage (or body-bias) randomization to "increase" the noise. The increment in

Table 2.1: Countermeasures against SCA

Countermeasure	Algorithmic	Architectural	Circuit	Transistor
Time	Masking	Shuffling, Dummy cycles insertion	Frequency Switching Multiple (or randomized) voltage (or body-bias domains)	Correlated time samples
Amplitude		Randomization, Hiding	Decoupling capacitors Shielding Current equalization	Equalization of current amplitude (e.g., dual-rail logic styles)

noise, in such cases, comes from embedding the information carrying `signal` in noise such that it "camouflages" it; although keeping in mind the performance and constraints. The methodology in all these cases consists of adding extra bits or additional circuity (i.e., at the cost of additional area and decreased performance)[126].

2.7.1 Transistor level Countermeasures

Dual-rail logic styles form the most common transistor level countermeasure which provides side channel protection in the amplitude domain. The inherent design seeks to equalize the power consumption levels irrespective of the switching transition. Common dual-rail logics are (but not limited to) SABL, WDDL, Dual-rail Precharge logic etc. To the best of our knowledge, there is no existing work which provides details of noise analysis on transistor level countermeasures. All of the above discussed countermeasures, which exist at the higher levels of design hierarchy, implicitly assume the workings of the lowest transistors, as it is, without accounting for the intrinsic physical noise coming from the MOSFETs. Consequently, very little design effort has been attributed to secure cryptographic circuits, as such from the bottom-most level.

2.7.2 Circuit Level Countermeasure

Circuit level countermeasures have showed a decent promise of camouflaging the signal by various circuit level techniques, such as voltage randomization. Existing literature offers a large scope of such work wherein using Voltage regulators. In addition, they also serve the purpose of energy efficiency by operating the cryptographic engine at a lower voltage compared to the auxiliary circuits, thus lowering the overall power consumption [56]. Such power management and voltage scaling (or randomizing) techniques have become more prevalent in the community over the past decade [12]. Coupled with other electronic techniques such as adaptive clocking, dynamic voltage and frequency scaling (DVFS),

as well as more common techniques such as clock and power gating, provide nevertheless, methods upon which the system performance can be improved upon as well as offering some side-channel security.

2.8 Randomizer based Countermeasures

In this section, we provide a brief survey of randomizer based countermeasures which employ the aforementioned techniques such as Voltage Regulation (VR), followed by a discussion of one of the performed works which motivated the exploration of IPN.

2.8.1 Related Works

VR based countermeasures, which rely on de-correlating the current at the supply voltage of interest, from the total load current is one of the ways to lower the overall signal amplitude. Switched capacitor based VR or switched-inductor based VRs comprise a linear voltage converter which converts the transient voltage, over a slowly changing time-constant load, clocked by an input signal. Low-dropout regulators (LDOs) are used to attenuate the high frequency components of the transient signal and reducing the overall bandwidth of set-up. Although, the authors in [110] provide impressive results based on simulations, their setup did not include any physical noise parameters.

As aforementioned, voltage scaling or voltage randomization is another technique wherein the voltage (of operation) is dynamically switched between different voltage levels through random voltage scaling or using switched capacitors. In [139], the authors propose to scramble the observed current by turning on and off the individual interleaved stages in a pseudo-random fashion. However, they do not provide the power/energy or area costs of their proposal and their security evaluation is uniquely based on power trace entropy, without indicating the amount of physical noise present in their evaluation system. The random voltage scaling technique proposed in [10] could lower the correlation coefficient by 10× when applied to the complete AES. Their approach aimed at FPGA designs and the authors suggested to alter the supply voltage once every 200 encryption rounds since in their settings, a successful DPA attack can be mounted against an AES engine after 2500 rounds and they need a changing supply rate much less than that.

Singh et al.[111] provided a more detailed review of the existing techniques and provide test measurement results on a 130nm chip. Their design demonstrated the usefulness of their approach of using a DVFS/R-FVD architecture using an Integrated Voltage Regulator coupled with a

digital modulation technique referred to as All-Digital Clock Modulation (ADCM), which requires considerably higher operating frequency clock (600 MHz). Additionally, they showed that SCA improvements were limited to a 9× less Measurements to Disclosure (MTD) when performing a CPA. Their results, impressive, nevertheless, do not provide any insight into the physical noise levels generated or needed for the effective working of such systems. From an information theoretic perspective, no clear indication is provided on the actual leakage which exists in such implementations.

Levi et al.[66] provide a case study combining signal hiding and algorithmic countermeasures and propose a simple randomizer which is able to provide comparable security levels when combined with masking. Further, they also propose a noise generation engine and through transient simulations validate the working of such a proposed oscillator with accumulated phase noise (jitter) [66]. The output of such a noisy oscillator is the passed on to shift registers in the design and subsequent gates.

2.8.2 Discussions from the Randomization Case Study

To begin with, we started with the V_{DD} randomizer case study [52] as a motivating work to understand the limits of IPN in low physical and high-physical noise regions, especially with reference to design and implementation of cryptographic implementations. To summarize briefly the work done in [52]

- An LDO-based Ultra-low Voltage (ULV) randomizer was proposed. The manifestation of the multiplicative noise is due to the modulation of the switching current from the LDO. The shorter transition times, which allow for multiple V_{DD} changes, make the profiling of the leakages difficult for an adversary. Furthermore, signal reduction, achieved through the usage of decoupling capacitors essentially filter and suppress the signal which has been shown to be effective across both low and high-physical noise regions, is achieved by protecting the circuit at ULV mode.

- An information theoretic approach [105, 114] was used to assess the security of the countermeasure since the IT metric is independent of the leakage model and allows for assessing the security across the whole physical noise region. From [30], the success rate of a maximum likelihood adversary is in proportion to the IT metrics employed.

- Although some promising results were made through the appli-

cation of the randomizer[13] but were limited to the to the HPNR or the boundaries of the low and high-physical noise region. Indeed, the authors appreciate the fact that generating noise in the LPNR, which is difficult to be exploited by an adversary, remains a challenge from a design perspective.

At this stage of the discussion, we briefly remind the reader about the variation of the noise with regards to the information leakage in the LPNR and HPNR.

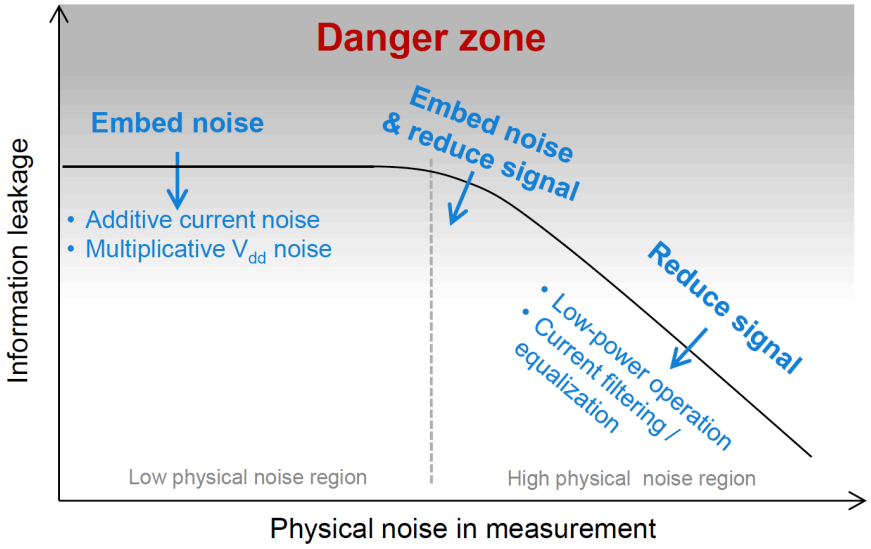


Fig. 2.10: Hardware Countermeasures against side-channel analysis [52]

Figure 2.10 provides a representation of noise *vis-a-vis* information leakage in cryptographic designs evaluations and attack scenarios. For lower physical noise levels, the information leakage from remains essentially constant; however, as the physical noise (both from the measurement and intrinsically) starts to increase, the amount of information leakage decreases. This is highlighted in the light-gray region of the curve. The dark-gray region of the curve, shows the areas where leakage is essentially higher and where the application of countermeasures does not provide any robustness.

[13] we refer the reader to [52] and do not mention here as they do not impact the thesis

Circuit level countermeasures typically work in the high physical noise (i.e., in the light-gray shaded area of Figure 2.10, marked by embed noise), by the addition (or rather embedding) noise such that the overall leakage of the system is "masked" by the presence of such noise. This noise, which could be a combination of both additive and multiplicative noise, is to be generated from the randomization circuits. Overall, this would lead to fall in the value of the SNR and consequently, the information leakage (measured by the PI). Figure 2.10 highlights two approaches,

1. Addition (or embedding) noise would work at the low-physical noise region interface.

2. At the higher physical noise region, signal reduction could (also) be a possible solution. However, it is of importance to understand that by higher physical noise, often physical noise due to measurement is implied, which is also the case in the aforementioned work. Intrinsic physical noise, on its own, has not yet been considered for this case study.

The key takeaways from the aforementioned work, which provided a basis for this thesis, could be summarized as below,

1. The characteristics of IPN in the Low-Physical Noise Regions (LPNR) form the vulnerable aspects of the V_{DD} randomizer design methodology, in which noise addition is favored using externally designed countermeasures. At Higher-Physical Noise Regions (HPNR) of operation, additional countermeasures (both circuit-level, such as LDO-based regulation, and mathematical, such as masking) suffice in reducing the information leakage from a leaking implementation. The LPNR remains a vulnerable area in terms of protection for cryptographic designers.

2. Additionally, it was shown that circuit level countermeasures are insufficient in guaranteeing the security levels desired, especially in the LPNR. It was also shown that such countermeasures are effective, once sufficient level of noise (especially in the LPNR), has been guaranteed. Further the cost of introducing such measures also needs to be accounted for; indeed, these lead to an increase in area and design complexity, as mentioned in [52].

3. Subsequently, this provided one of the major motivations to analyze the intrinsic physical noises of devices; understanding their behavior was critical in alleviating the effect of circuit or such system level countermeasures, while at the same time since the inherent noise

from the same transistors are being utilized, the area cost and design complexity are not increased (atleast at the lowest transistor levels) and the gains amortized. This lead to the major contributions of this dissertation work.

4. Comparing different countermeasures across different levels of abstractions (e.g., circuit level, transistor level or the system/protocol level) is not an easy task on account of the different noise levels added at each level; nevertheless, the above work [52] could provide a relative guideline with which to compare the total noise evaluated from an implementation. However, strictly following such a defined level is *highly discouraged* as differences in the respective implementations viz. simulations v/s real-world implementations, filtering setup, noise injected due to additional components which may or may not have been included in the simulation setup[14] and difference in the technology levels (viz.65nm and 28nm technology nodes) are some of the factors which should be carefully considered when comparing such results across such varying levels of implementations.

2.9 Recap Summary

Although the introductory chapter does not require a Conclusions section at its end, we very briefly re-iterate the key takeaways to recollect the reader about the material covered.

1. Introduced the necessary tools and evaluation metrics (both from a physics and a cryptographic perspective).

2. Provided details about the device technology, power consumption models for side-channel leakage modeling, including details on the algorithms and hardware designs used in the dissertation.

3. Provided didactic approaches on the simulation tools and methods used in this thesis work.

4. Provided the motivation for the role of IPN by highlighting the issues encountered by circuit level countermeasures, especially for low-physical noise regions.

[14] although care has been taken to accurately model the simulational setup (ref. Chapter 6) as close as possible to the setup provided in [52], nevertheless the random number generation part and the noise injected due to the LDO was not included to minimize duplication and to allow for a setup based on generic measurement for SCA rather than for following a specific setup

Chapter 3

Scaling Trends Analysis

In this chapter we analyze the scaling trends of dual-rail logic styles w.r.t CMOS implementations and its impact on side-channel analysis. We look at the technology scaling of devices from 65 nm *to 28* nm *and the corresponding impact on the cryptographic figures-of-merit. We conclude by motivating the reader the necessity of designing and exploring noisy CMOS implementations.*

3.1 Motivation

Dual-rail logic styles have been considered as possible alternatives to CMOS for the design of cryptographic circuits (more) secure against side-channel attacks. The state-of-the-art view on this approach is contrasted as they reduce the exploitable side-channel signal while not being sufficient to fully prevent the attacks. Since the limitations of dual-rail logic styles are essentially due to implementation challenges (e.g. the need of well-balanced capacitances), a natural question is to find out how they evolve with technology scaling [119]. Here, we discuss this issue based on the relevant case study of an AES S-box implemented in CMOS and a dual-rail logic style, for two (65nm and 28nm) technologies. Our evaluations show that the security vs. performance trade-off of our dual-rail logic style does not scale well compared to CMOS. It also shows that the scaling trends for CMOS are more positive (i.e. smaller technologies and supply voltages reduce the energy consumption and the side-channel signal). So these results suggest that dual-rail logic style may not be a sustainable approach for side-channel signal reduction as we move towards lower technology nodes.

3.2 Existing State-of-the-Art

Following the first publications on power and electromagnetic analysis against cryptographic implementations, dual-rail (aka dynamic and differential) logic styles appeared as promising candidates to improve security against such attacks. Intuitively, these logic styles aim to solve the issue directly at the circuit level, by trying to reduce the side-channel Signal-to-Noise Ratio (SNR). For this purpose, they typically ensure that the switching activity of the circuits is independent of the manipulated data. However, despite constant switching activity, small data-dependent variations in the current traces can generally be observed, e.g. due to the unbalanced capacitances of the gates differential nodes and their interconnections. Therefore, a large body of work investigated the design of dual-rail logic styles in order to reach the best security vs. performance tradeoff, including but not limited to SABL [121], WDDL [123], DyCML [19], MCML [28] and MDPL [97]. Furthermore, DPL based designs suffer from the flaw of "early evaluation", which is an architectural imbalance created by different paths (hence causing different delays) at the gate inputs. To avoid this flaw, improved DPL was devised to synchronize the inputs at the gates (iMDPL [92], BCDL [85] etc.) similar to the operation in asynchronous logic. Evaluations based on both simulations and actual measurements then confirmed that getting rid of these data dependencies is extremely challenging [122, 69, 96, 102]. More recent works even showed that filtering effects in concrete measurement setups make these small data dependencies reasonably easy-to-exploit, e.g. thanks to linear regression [54]. To complete the picture, most of these dual-rail logic styles usually come with significant performance overheads, some of them additionally requiring full custom-design (which allows further control of the hardware, but is making development and deployment significantly more challenging/expensive).

In parallel, recent progresses have shown that mathematical countermeasures against side-channel analysis, and in particular the mainstream shuffling and masking techniques [23] [43] [73], can only lead to significant security improvements if the side-channel SNR has been sufficiently reduced beforehand [115] [134]. This raises the problem of finding effective (and if possible efficient) hardware techniques allowing to fulfill this condition. Intuitively, it can be done by reducing the side-channel signal, which is what dual-rail logic styles achieve, or by increasing the noise.

Eventually, since the evaluation of secure hardware technologies goes together with technology scaling, another problem is to find out which of those approaches has more potential for the future.

3.3 Main Contributions

In this chapter, we investigate the impact of technology scaling on the security (and performance) trends of CMOS and Dual-rail logic implementations. Our contributions are as follows:

1. Analyzed the impact of technological scaling, both in terms of performance metrics and security evaluations, for CMOS and DDSLL based circuits using simulations and showed how dual rail logic styles (such as DDSLL) lose interest (in terms of security) with scaling.

2. To achieve consistency across different supply voltages, we depict a complete trend of both analysis (i.e., performance and security trends) across the working range of supply voltages as allowed by the technology nodes under consideration.

3. As a takeaway from this work, we provide the necessary motivation behind the choice of design styles for the implementations and the necessity of analyzing IPN going forward, in this thesis work.

3.4 Preliminaries

Although we have covered the preliminaries and necessary background in Chapter 2 of this work, we revisit this solely in the context of dual-rail logics, which are studied in this chapter. We briefly recollect the power models used in CMOS logic style to familiarize the reader with the power model used for dual-rail logics (DDSLL in this case). We also, present a circuit level explanation of the basic digital building logic of such dual rail logics, namely, a generic DDSLL gate.

3.4.1 Logic styles: CMOS and DDSLL

Traditional CMOS circuits have shown a data-dependency in the power consumption leading to exploitable side-channel information, e.g. thanks to Differential Power Analysis (DPA) [63], Correlation Power Analysis (CPA) [20] and Template attacks [24]. The power consumption of a CMOS circuit is modeled by the following equation:

$$\begin{aligned} P &= P_{dyn} + P_{stat}, \\ &= \frac{1}{2} N_{nodes}\, \alpha_F\, C_L\, V_{DD}^2\, f_{clk} + I_{leak}\, V_{DD}, \end{aligned} \tag{3.1}$$

where P_{dyn} is the dynamic power consumption, P_{stat} the static power consumption, N_{nodes} is the number of nodes in the circuit, α_F represents

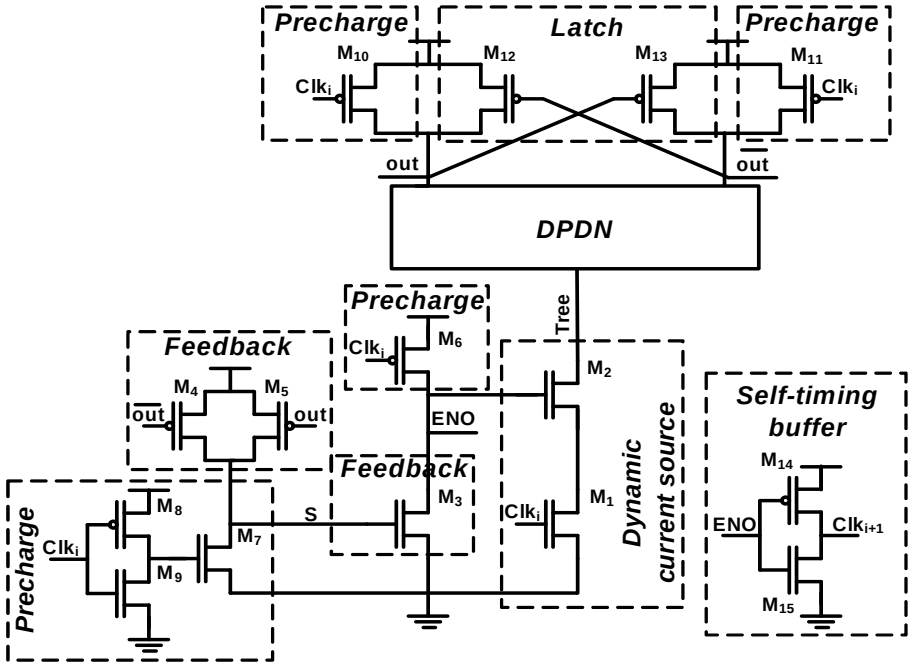


Fig. 3.1: Schematic of a generic DDSLL gate [53].

the activity factor of the design, C_L is the load capacitance, V_{DD} is the supply voltage, f_{clk} represents the clock frequency and I_{leak} denotes the leakage current. We assume here that one operation is executed per clock cycle, and thus $f_{clk} = f_{op}$, which determines the target throughput of the application. The operation period $T_{op} = \frac{1}{f_{op}}$ should be more than the critical path delay T_{del} to guarantee correct functionality. The data-dependency of the CMOS logic comes from both its dynamic and static power consumptions. In the dynamic part, α_F directly depends on the data being processed. In the static part, the I_{leak} is the data-dependent parameter. Although the former is dominant, static power consumption can also be exploited if its value is sufficiently high and the operating frequency is low enough allowing the reduction of noise via simple averaging techniques [26, 81].[1] Yet, in our following experiments, dynamic power indeed dominates.

In comparison with CMOS, we investigate the Dynamic and Differential Swing-Limited Logic (DDSLL) [121] which aims at low-power

[1]Note that advanced technologies usually provide multiple flavors such as low-power and high-performance along with different device choices such as high and low threshold voltages, providing circuit designers with various options to reduce the power consumption – and the leakage power as well – which may modify the respective importance of these source of leakages.

implementations and of which the dynamic power consumption is given by:

$$P_{dyn} = \frac{1}{2} N_{nodes} \alpha_F C_L V_{DD} V_{swing} f_{clk}, \quad (3.2)$$

where α_F equals 1 because all dynamic and differential logic styles ensure one output transition per clock cycle (independent of the data being processed), and V_{swing} is the output voltage swing. DDSLL gates are designed to have limited swing (i.e. $< V_{DD}$) in order to reduce the dynamic power consumption and hence the energy per operation.

Figure 3.1 shows the circuit of a generic DDSLL gate. A Differential Pull-Down Network (DPDN) is used to evaluate the required function. It mainly consists of NMOS transistors. The DDSLL gate employs a dynamic current source to significantly reduce the static power consumption similar to what is achieved in the DyCML logic style. The cut-off of this current source is performed via a feed-back network which signals the end of an operation, so that the self-timing buffer creates a clock signal Clk_{i+1} declaring the termination of the current evaluation phase. This clock signal is used to feed the following DDSLL gate. The precharge transistors are used to precharge the differential outputs of a DDSLL gate to the supply voltage before an evaluation of the gate's function takes place, and the latch transistors preserve the evaluated voltage at the differential output nodes.

The operation of a DDSLL gate is quite simple. It works in two modes: precharge and evaluation. In the precharge mode, when the clock signal Clk_i is low, the outputs *out* and $\overline{out}$ are precharged to V_{DD}. There is no current path from V_{DD} to GND because transistor M_1 is switched off. However, transistors M_6 and M_2 are switched on. Next during the evaluation phase, Clk_i goes high turning on the transistor M_1 while M_2 is still on (both forming the dynamic current source) as node ENO was previously charged to V_{DD} in the previous precharge phase, thus creating a path to discharge one of the output nodes to *GND*. The discharge path through the DPDN network is the one with the lowest impedance depending on the inputs being processed. As one of the outputs falls below the threshold voltage of the feed-back transistors (M_4, M_5), one of them will turn on which in turn will discharge the node ENO to GND thus starving the current source. The design of a DDSLL circuit comprising of several functions can benefit from resource sharing (the dynamic current source, parts of the precharge circuit, the feedback circuit and the self-timing buffer are functions that evaluate at the same time), therefore reducing the area cost and the overall power consumption.

In all differential design styles, the unbalanced capacitances are

considered as a source of information leakage. They mainly come from either routing imbalances or from internal imbalances. In this paper, we only consider the latter ones. (As mentioned in section 3.5.1, additionally considering post-layout simulations or variability should amplify the trend we put forward. The process and architectural mismatches, if considered, would only exacerbate the imbalance and hence result in an even wider gap). Hence, our choice represents a sort-of best-case scenario already. In this respect, we note that the only way to eliminate internal imbalances is to design circuits with perfectly symmetrical differential gates, which is usually very expensive in terms of area and speed. Hence, our DPDN designs exploit the binary decision diagram technique from [38] in order to improve performances by exploiting more complex gates with minimum imbalance caused by internal capacitances (details can be found in [53]).

3.4.2 Evaluation metrics

Evaluating logic styles across technology scaling and supply voltage scaling is a challenging task, since certain metrics may favor one logic style over another. In order to be as fair as possible in our performance and security evaluations, we therefore selected generic metrics that are generally more reflective of the "global performance level" of an implementation, and can capture any type of information leakage.

More precisely, and as far as performances are concerned, the energy per operation is a quite discriminant[2] metric, as it corresponds to an integral over time, and therefore is not "compressible" (via architectural tweaks) beyond what is allowed by the total combinatorial cost of an implementation [58]. Concretely, the energy consumption of a logic circuit can be calculated by integrating the power consumption over the time required for the target operation (in our case-study, the AES S-box):

$$\begin{aligned} E_{op} &= \int_t (P_{dyn} + P_{stat})\, dt, \\ &= \underbrace{\frac{1}{2} N_{sw}\, C_L\, V_{DD}\, V_{swing}}_{\text{Dynamic}} + \underbrace{V_{DD}\, I_{leak}\, T_{del}}_{\text{static}}, \end{aligned} \tag{3.3}$$

where $N_{sw} = \alpha_F N_{nodes}$ is the number of switching nodes for the operation and T_{del} is the circuit delay (following the investigations in [17]). In the

[2] As aforementioned, evaluation metrics should be independent of effects such as technology node scaling or supply voltage variations and should be able to discriminate across such designs without any bias

case of CMOS logic style, V_{swing} is equal to the supply voltage being used.

A note on the Power-Delay Product (PDP) Although the above section clearly identifies the E_{op} as the choice of evaluation metric, the Power-delay-Product (PDP) is another similar metric used when comparing implementations. Though not very granular as the E_{op}, the PDP nevertheless, offers a quicker way of evaluating the energy metrics of the design and offers interesting points of discussion. Since the PDP is defined at the gate level, compared to the E_{op} which is defined at the bit-level, it predicts the average energy consumed per switching event (i.e., for either a $0 \rightarrow 1$ or a $1 \rightarrow 0$ transition). We define the PDP as,

$$PDP = C_L V_{\mathrm{DD}}^2 f_{max} t_p = \frac{C_L V_{\mathrm{DD}}^2}{2} \tag{3.4}$$

where, assuming the gate switches at full speed,

$$f_{max} = \frac{1}{2t_p} \tag{3.5}$$

Security Evaluation Metrics As for the security metrics, our choice was dictated by various constraints and features. First, the shape of the instantaneous dynamic power (i.e. the side-channel signal) changes significantly depending on the technology and the supply voltage used. Therefore, it is important to consider these changes while evaluating the CMOS and DDSLL logic styles, and to consider a multivariate analysis. Second, our target implementations are not masked, and therefore our simulations exclusively exploit first-order leakages. So while in general, a fair comparison of our logic styles would require to compute a mutual information metric [114], in this particular case we can simplify our evaluations by (*i*) applying a dimensionality reduction to our traces, namely a Principal Component Analysis (PCA) which will capture the shape of the noise-free simulated traces [7], and (*ii*) computing Mangard's SNR on the reduced traces [72], which is equivalent to the mutual information metric in this case [74, 29].

For this purpose, we denote a power trace as $\boldsymbol{l}$, its corresponding random variable as $\boldsymbol{L}$ and assume that this random variable is a function of the S-box input X, and a noise random variable $\mathcal{N}(\mu, \sigma^2)$. The multivariate traces (ref. Equation 3.10) are reduced to univariate ones thanks to PCA (ref.Equation 3.11). Giving the trace l a subscript x corresponding to the input and a superscript i corresponding to an

index (since the trace for a plaintext x can be measured multiple times), Mangard's SNR is defined in Equation 2.30. Mathematically,

$$L = X + \cancelto{0}{\mathcal{N}(\mu, \sigma^2)} \tag{3.6}$$

$$L = X \tag{3.7}$$

$$X = \left\{x_1, x_2, \cdots, x_i\right\} \tag{3.8}$$

$$\boldsymbol{l} = \Pr(X|L) \tag{3.9}$$

$$\boldsymbol{l} = \left\{l_1, l_2, \cdots, l_i\right\} \tag{3.10}$$

$$l = \mathsf{PCA}(\boldsymbol{l_i}) \tag{3.11}$$

$$\max_i \left\{ \hat{\mathsf{var}}_x(\hat{\mathsf{E}}_i(L_x^i)) \right\} \tag{3.12}$$

In our following simulations, this SNR will be computed for noise-free traces (ref.Equation 3.6). This amounts to maximizing the signal (ref.Equation 3.12) (i.e., we ignore the denominator of Equation 2.30 in this case).

Note that for readability, our following results only report the quantification of our experiments with this security metric. However, various other (heuristic) choices could be considered, e.g. computing the SNR for the most informative samples in the traces, or considering more dimensions after the application of the PCA. In our study, none of these variations (that we also browsed) lead to different conclusions regarding the two main trends outlined in introduction.

3.5 Target designs

For the sake of simplicity, and in order to demonstrate the technology trends of CMOS and DDSLL, we chose an 8-bit AES S-box as the benchmark circuit as described in section 2.5.1. The CMOS and DDSLL AES S-boxes were implemented in a full-custom fashion, using the CADENCE Virtuoso tools, in a low-power 65 nm bulk technology and 28 nm FDSOI technology.

The gate design of CMOS and DDSLL S-boxes in both 65 nm bulk and 28 nm FDSOI technologies are kept identical, i.e. the gates used and the number of transistors remain unchanged. However, we respected the minimum feature size of each technology (to decrease the switching capacitance, hence the energy per operation) and resized the transistors' widths adequately to guarantee functionality under the given operating conditions (the clock frequency of operation and the overall voltage

operating range). In 65 nm bulk technology, standard threshold voltage (SVT) transistors are used to reduce the static current while maintaining good performances with respect to the circuit delay. For benchmarking purposes, we also implemented the CMOS and DDSLL S-boxes using the low threshold voltage (LVT) transistors available from the same technology. In 28 nm FDSOI technology, both SVT and LVT transistors were again used to maintain a fair comparison with the 65 nm technology. Yet, for readability, our following results only report the quantification of our experiments with SVT transistors (trends are again identical for LVT transistors). Eventually, the widths of the DDSLL S-box transistors were chosen such that the voltage swing is sufficient for the circuit to operate correctly at the lower limit of the supply voltages of each technology.

In our experiments, all the S-boxes are fed with buffered inputs to maintain equal fan-ins and have realistic inputs, yet the DDSLL S-box additionally has a buffered clock. Also, all the outputs of the S-boxes are loaded with equal fan-out buffers. Each S-box is provided with a separate supply voltage than that of the input/output buffers so that the buffers' energy consumptions are not taken into account in our evaluations.

3.5.1 Simulation settings

Simulations for the above designs are done at the schematic level (without any extracted post-layout capacitances) using Eldo simulator based on SPICE models provided by the industrial foundries at room temperature of 25°C.

A Note on Process and Architectural Mismatch In this respect, we note that any imbalance in the parasitic elements would affect the difference between the delay of the differential routes of the DDSLL S-box, which would impact its power consumption, leading to a higher (exploitable) signal being observed by practical attacks [104]. And this is expected to get only worse with technology scaling and variability [8], since balancing the capacitances in an implementation [94] naturally becomes more challenging with smaller circuits and smaller routing capacitances. Therefore, and as previously mentioned, taking the parasitic routing capacitances into consideration could only amplify our observation (which is that the comparative advantage of DDSLL over CMOS is vanishing with technology scaling). A similar statement holds for other effects that we did not take into consideration in this work such as crosstalk and the influence of process, voltage and temperature (PVT) variations.

Consequently, we assume our results correspond to an ideal scenario and the inclusion of more physical defect (or imperfections)/(s) should only deteriorate the performance of dual-rail logic styles compared to CMOS.

The frequency of operation is chosen to be 10 MHz, which is in accordance with the usual operating frequencies for cryptographic applications (see, e.g.[11]). The supply voltage is swept across a range of 500mV, in steps of 100mV starting from the nominal voltage of each technology, namely, 1.2V and 1V for the 65nm bulk and the 28nm FDSOI technologies, respectively. The lower limit of the supply voltages is imposed by the correct functionality of the circuit at the target frequency for a given implementation.

To calculate the energy per operation, we considered 1000 random input signals. As for the security analysis, the S-box input signal has 256 possible values whose transitions are chosen between 0 and an arbitrary input. Restricting the inputs to a subset from the 256^2 possible inputs was mainly motivated by practical simulation constraints (memory and simulation time) and is not expected to strongly impact the comparison between the logic styles.

A Discussion on the V_{th} and Sub-threshold Design Although sub-threshold designs are commonly favored in the design of electronic systems towards lower power and lower-energy systems, their application in a cryptographic context needs to be re-analyzed in view of the application. Since hardware-based cryptographic devices are generally, part of a hardware-software (HW/SW) based co-design approach, and the HW component serves more an accelerating/additional component, sub-threshold designs could very well impact the performance of such systems. Since SW-based systems tend to operate at higher voltages (for instance, a microcontroller based system could well operate at 0.8 to 1.2V), the equivalent HW components need to keep up with the same (if not more) speeds to maintain coherency. However for certain applications, for instance, such as the ones employed in light-weight implementations, could indeed very well exploit a sub-threshold design based approach as such applications are not traditionally power intensive or performance critical.

Note that operating at higher supply voltages and using the 28nm technology node allows the circuit implementation to run at higher frequencies. But the frequency choice has no impact on our results since we have chosen to use the energy per operation as the evaluation metric and not the total power consumption.

3.6 Comparative scaling trends: CMOS vs. DDSLL

In this section, we aim to compare the CMOS and DDSLL logic styles and study how their security versus performance tradeoff evolves with technology and supply voltage scaling. To be able to do that in a comprehensible manner, we plot the ratios between our (security and performance) metrics computed for both CMOS and DDSLL, one in function of the other.

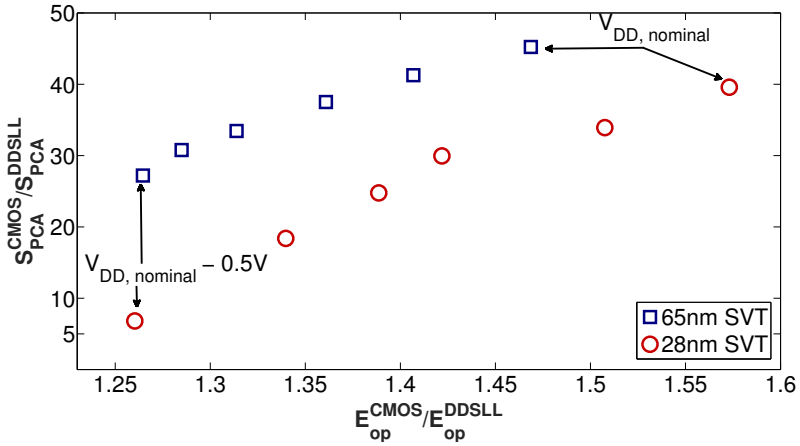


Fig. 3.2: Evolution of the tradeoff between the PCA signal ratio and the E_{op} ratio for CMOS vs. DDSLL using 65 & 28nm technology nodes.

More precisely, Figure 3.2 shows the PCA signal ratio between CMOS and DDSLL S-boxes versus the energy per operation ratio between these logic styles for the 65 nm bulk and the 28 nm FDSOI technologies (designed using SVT devices). The different points represent the supply voltages we used that span over a range of $0.5V$ starting from the nominal supply of each technology ($1.2V$ and $1V$ for the 65 and 28 nm technologies, respectively). This study allows us to make the following observations:

1 By reducing the supply voltage, the energy per operation ratio of CMOS with respect to DDSLL is decreasing for the 65 and 28 nm technologies. This reduction can be explained by the fact that the E_{op} value of the DDSLL style decreases almost linearly with the supply voltage, because it maintains nearly the same voltage swing while E_{op} of CMOS decreases quadratically (see Equation 3.3).

2 As for the PCA signal ratio between CMOS and DDSLL (for both technologies), it also decreases with the supply voltage scaling. Again, this is due to the fact that the voltage swing of the DDSLL

S-box is kept almost unchanged leading to a slow reduction rate of the transient power consumption. Hence the PCA signal with V_{DD} scaling is less compared to that of CMOS.

3 Most importantly, Figure 3.2 illustrates that at 28 nm technology, the PCA signal ratio between CMOS and DDSLL is less than that of the 65 nm technology for similar E_{op} ratios. This figure neatly puts forward that the security vs. performance tradeoff between these logic styles does not scale positively for DDSLL, even though we do not consider any routing parasitics or PVT variations in our simulations.

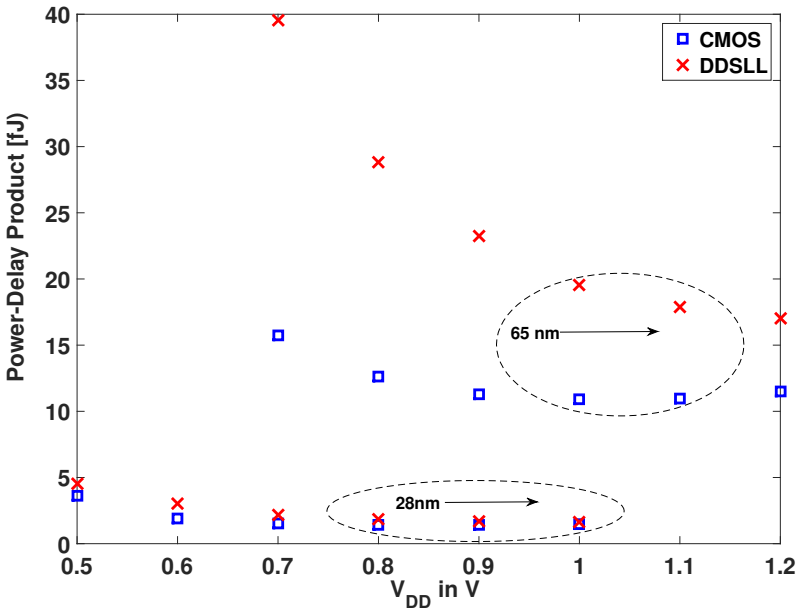


Fig. 3.3: Evolution of the PDP for CMOS vs. DDSLL using 65 & 28nm technology nodes.

Figure 3.3 depicts the PDP for the CMOS and DDSLL S-box designs across 65nm and 28nm nodes. We discuss the following,

1. 65nm based curves, shown towards the more right side of the graph, clearly predict higher values of the PDP, when compared to 28nm designs. Naturally, this is to be expected as the output capacitances of 28nm are comparatively lower to the 65nm technology.

2. For both the technology nodes, for the DDSLL designs, depict higher values of the PDP when operating just above their respective

threshold voltage (we denote this voltage as $V_{\text{th}} + \delta$) values compared to their CMOS counterparts at the same points-of-interest; i.e.,

$$PDP(28\text{nm}, 65\text{nm})|^{CMOS}_{(V_{\text{th}}+\delta\text{V})} < PDP(28\text{nm}, 65\text{nm})|^{DDSLL}_{(V_{\text{th}}+\delta\text{V})} \quad (3.13)$$

3. From Figure 3.3, it is also clear that the gap between CMOS (indicated in blue curves) and DDSLL (indicated in red red) has shrank when moving from 65nm to 28nm. This validates our earlier conclusion and furthermore leads to the observation that dual-rail technologies do not hold any additional advantage (solely in terms of electrical performance metrics such as the PDP) when moving to lower technology nodes. To summarize,

$$\begin{aligned} |PDP_{DDSLL} - PDP_{CMOS}||_{28\text{nm}} < \\ |PDP_{DDSLL} - PDP_{CMOS}||_{65\text{nm}} \end{aligned} \quad (3.14)$$

It is worth emphasizing that similar observations were made by comparing CMOS to DDSLL S-boxes using LVT devices in both technologies. Therefore, changing the device type leads only to either better performance or more power savings, but the technology and supply voltage scaling trends remain the same. Also, and for the sake of completeness, we conducted the same experiments using the maximum SNR (before PCA) as a security metric and the technology and supply voltage scaling trends remained unchanged.

Clarification regarding the usage of PCA As previously mentioned in section 2.5.5, the Principal Component Analysis (PCA) is a method by which the information contained in a trace can be filtered/reduced such that only the most informative contents are retained and the remaining may be disregarded to reduce the dimensionality of the traces and facilitate quicker post-processing. As emphasized in the previous paragraph, the technology and supply voltage trends remains unchanged irrespective of the application of PCA on the original traces, within the scope of our experiments.

In addition, we note that changing the frequency of operation does not impact our conclusions as long as the dynamic power consumption dominates. We simulated the S-boxes down to 100kHz and the technology and voltage scaling trends were again the same.

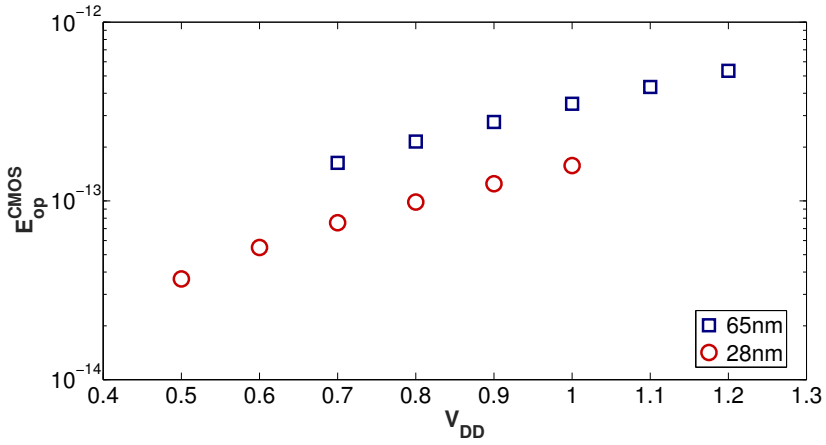
Eventually, we note that technology scaling is advancing at a fast pace and secure implementations will soon follow (given the fact that up until now applications such as smart cards tend to lag by one or

two technologies). Also, circuit designers generally aim at scaling the supply voltage in order to further reduce the energy consumption of the digital circuits (sometimes operating below the transistor sub-threshold voltage leading to minimum energy per operation). Both trends lead us to conclude that the observations in this section may rapidly have concrete relevance.

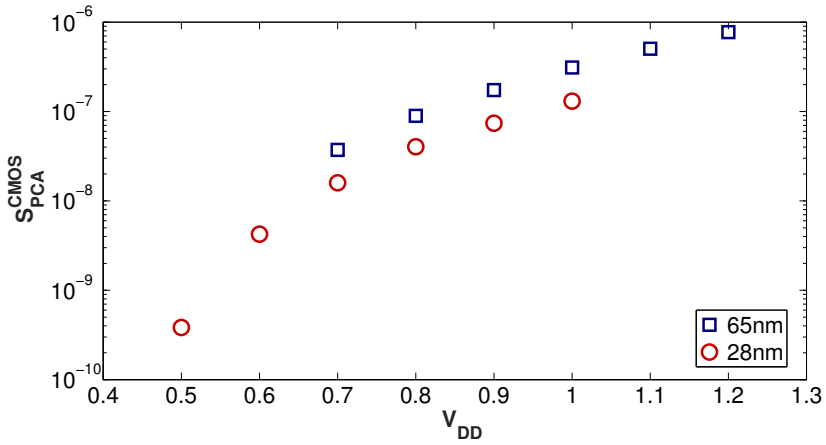
3.7 Technology and V_{DD} scaling trends for CMOS

Since the comparative advantage of DDSLL over CMOS vanishes in our case study, one natural complementary question is whether scaling trends for CMOS circuits lead to a more positive conclusion. In this section, we answer this question by focusing on the impact of technology and V_{DD} scaling on CMOS circuits only. For this purpose, Figure 3.4(a) reports the energy per operation of the CMOS S-box for different supply voltages, using both 65 and 28 nm technologies. We recall that the minimum V_{DD} was chosen such that the CMOS S-box operates correctly at the 10MHz target frequency for each technology. As expected, the energy per operation of CMOS decreases almost quadratically with the reduction of the supply voltage (see Equation 3.3). The figure also shows clearly that technology scaling from 65 to 28nm reduces the energy per operation (of the CMOS S-box) by a factor of $2.2\times$. This reduction is compliant with the expected technology scaling trend as explained in [42].

Similarly, Figure 3.4(b) shows the signal after PCA of the CMOS S-box for different V_{DD} values, using both 65 and 28nm technologies. The PCA signal of CMOS decreases as the supply voltage scales down. In the 65nm technology, the S_{PCA} reduction is more than one order of magnitude across the whole V_{DD} range ($1.2V$ to $0.7V$) and it reaches even more than two orders of magnitude in the 28nm technology, by reducing the supply voltage from $1V$ down to $0.5V$. As for the technology scaling from 65 to 28nm, it is also clear that the signal after PCA of the CMOS S-box decreases by a factor of $2.3\times$ at comparable supply voltages. So the scaling trends for PCA signal is also positive for CMOS. Here, we note that an analytical explanation of these observations is more challenging, since the small data-dependent current variations that lead to exploitable side-channel signal are much harder to capture theoretically. (Yet, we can suppose that the aforementioned signal reductions essentially originate from the interactions between reductions of the current and variations of the load capacitance). Essentially, we have shown that (internal) imbalances have an increasing impact with technology scaling.



(a)



(b)

Fig. 3.4: Scaling trends of E_{op} and the PCA signal for the CMOS S-box across a supply voltage range of $500mV$ for 65 and 28nm tech.

A note on the body-biasing in FDSOI Fully Depleted Semiconductor-On-Insulator (FDSOI) technology allows for the biasing of the back gate (i.e., the bulk gate of a MOSFET device). This allows for fine tunability of the threshold voltage V_{th} depending upon the application (i.e., either for low-power (RVT devices) or high-performance (LVT devices)) and better control of the Gate-Induced Drain Leakage (GIDL). Depending upon the type of biasing applied, we have Reverse Body/Back-gate Bias (RBB) (reverse back-gate or body bias) wherein a negative voltage is applied to the back-gate of the transistor, reducing the leakage current

although at the cost of lowered performance. Compared to Forward Body/Back-gate Bias (FBB) (Forward body or back-gate bias), a larger range of voltage is applied which allows for drawing a higher current and increased performance, at the cost of higher leakage.

When considering the design of side-channel secure cryptographic circuits, and especially within the scope of this thesis, the default values of the body/back-gate bias were fixed for the corresponding NMOS (i.e., 0V applied at the body-gate) and PMOS (V_{DD}V applied at the body gate) transistors. This was done so as to not perturb the noise PSD (obtained via either `.AC` or `.NOISETRAN` simulations; since they are bias-dependent which varies with any change in the V_{th}). However, we (already) noted the importance of modulating the back gate as a possible countermeasure and recent work has validated the same. In a very recent work [25], the authors have demonstrated that by applying back-gate bias in the forward direction led to a increased DPA resistance of $\times$14.5. The authors were able to randomly change the back-gate of the bias (effectively increasing the noise of the implementation). Overall, the notion is the randomization of the back-gate voltages such as to increase the overall noise of the system (since the increased leakage current, thanks to the biasing, is considered noise for Power Analysis Attacks (PAA) type-of attacks (such as the Differential Power Analysis (DPA)) and hence will build up better resistance [25]). The authors demonstrated that by applying back-gate bias in the forward direction led to a increased DPA resistance of $\times$14.5. This leads to an apparent observation, i.e., the body-biasing could indeed be used an effective countermeasure against SCA and offers a scope of interesting future research.

3.8 Conclusions and Future Directions

In this case study, we analyzed for the first time the comparative scaling trends of CMOS and dual-rail logic styles. In short, our evaluations suggest that the interest of DDSLL over CMOS, expressed in terms of a security vs. performance trade-off, vanishes as circuit sizes shrink. To a good extent, we believe a similar conclusion should be obtained for other dual-rail logic styles. In particular, from the security point-of-view, they all suffer from capacitance imbalances to some extent, and this phenomenon can only be magnified in smaller technologies. While our case study was based on an AES S-box, we believe similar trends should also be obtained for full AES implementations. Indeed, similar energy trends will be integrated over more clock cycles, and the signal variations exploited in a DPA are anyway focused on the first cipher rounds.

On the need of IPN and its usefulness These results therefore suggest that reducing the SNR in advanced technologies may be better achieved by accounting for the intrinsically occurring noise levels than by reducing the signal using dual-rail logic styles. It also suggests the design of `noisy` and efficient CMOS implementations as an interesting scope for further research. Since IPN is (theoretically) not dependent on the capacitative imbalance between, for instance, differential rails, it is worth exploring in this context.

We finally note that while dual-rail logic styles may not be a sustainable solution for signal reduction purposes, it remains possible that they are helpful ingredients of physically secure implementations for other reasons (e.g. in order to facilitate the independence condition that is required for masking to deliver its security promises [137]); Levi et al. in [66] show how such signal reduction hiding techniques can indeed be combined with dual-rail logic styles (WDDL in their case) and additionally utilize a noise generation engine to reduce the overall SNR and improve the security of a masked leaking implementation. Exploratory research into the integration of novel logic styles combining circuit/physical level noise-based countermeasures remains indeed an interesting scope for further research.

Chapter 4

Transient Noise Analysis - I

In this chapter we introduce our design methodology for evaluating and quantifying the intrinsic device physical noise of a cryptographic implementation. We further explore how the simulation parameters can be optimally chosen for the qualification into the desired methodology. We present a mathematical approach detailing the introduced methodology and provide an assessment of the trade-offs in terms of the simulational cost and complexity.

4.1 Motivation

Side-channel attacks, such as differential power analysis (DPA) [63], exploit physical-leakage signals from a cryptographic device to extract sensitive information (*for e.g.,* cryptographic keys). Circuit logic-styles such as dual-rail [82], have been proposed to reduce the signal within the leakage. However, with technology scaling their (leakage) signal reduction is hindered by increased capacitance imbalances compared to standard CMOS logic styles [88] (*e.g.,* moving from 65nm to 28nm nodes). Existing countermeasures against side-channel analysis such as shuffling (adding noise in time domain) and masking (or algorithmic noise, adding noise in the amplitude domain) work well in scenarios where the SNR has been sufficiently reduced. Exploring how to increase the intrinsic physical noise coming from the transistors themselves or its security extent (to reduce the SNR) is very important. Mainly due to the fact that such noise element is inherent (*intrinsic*) to the device itself, and it is quite challenging for an adversary to eliminate/reduce it. In this chapter, we explore the design of *noisy* CMOS implementations, and propose a methodology to accurately evaluate the *intrinsic* physical MOSFET

noise allowing designers to derive insights and understanding of its impact through gate-level simulations. More specifically, we investigate a methodology to answer the following questions, *i.e.,* can we derive a concrete methodology to link inherent MOSFET noise to well established cryptographic merits, how does the noise (and its security implications) scale with the number of transistors/devices or their sizes, and how does it behave as a function of the voltage supply (for e.g, a common side-channel attack scenario in which adversaries lower the supply voltage).

4.2 TIMESPAN : A Simulation Methodology

Conventional noise analyses in analog and RF circuit designs mostly use the ac or the harmonic based approaches. However, in digital cryptographic applications, where each point in a transient run is potentially a source of information leakage (from an adversary perspective), it makes it worth analyzing the effect of noise on *each* time sample. From a cryptographic perspective, this is a univariate analysis compared to a multivariate analysis (where multiple time-samples are used). In the scope of this work, we focus on the univariate aspect only. The time sample with the highest value of signal (or SNR) is chosen as the point-of-interest (POI), as this usually represents the worst case for defending against a side-channel adversary. Using the transient noise simulation tool in Eldo circuit-simulator software (provided by Mentor Graphics) [78], we provide a methodology to evaluate the *intrinsic* physical noise sources (i.e., noise coming from the transistors themselves and *not* externally) and calculate the resulting "cryptographic" signal and noise (units of A^2) as defined in (2.30), and compare them to the values obtained from ac simulations and variance calculations. Our introduced methodology is not limited to the designers of cryptographic implementations; in fact they can be extended to evaluators in a cryptographic setting who wish to evaluate the intrinsic side channel resistance of an implementation and building upon to add external countermeasures. We refer to our introduced methodology as TIMESPAN: A Transient Noise Analysis Methodology for Evaluating the Intrinsic Physical Noise of Cryptographic Designs, highlighting the importance of *transient* behavior of the IPN.

Although noise by itself is a random process which is naturally frequency dependent, in the context of this thesis, we explore the noise dependency on frequency through the `.AC` analysis and subsequent PSD-analyses of the transient simulated noise traces by post-processing. Indeed, the noise content (from a cryptographic perspective) could also be explored through the frequency spectrum (with additional filtering

caveats), we nevertheless include such treatment at relevant places as possible. However, since analog (or digital) simulations are inherently mixed-signal and almost always *transient* in nature, it is natural to explore the transient behavior of such devices. The frequency aspect aids in furthering our understanding of such behavior in a highly complementary fashion. Nevertheless, we do not rule out the scope of analyzing such behaviors in a more frequency-based context and hypothesize that such works could indeed by useful for analyzing more complex multivariate distributions where the frequency dependence of the noise on a given informative signal component is not the exclusive sum of a single component but rather the sum of a more complex (harmonics) of different frequencies. Such analyses, from a research perspective, could lead to intuitions in understanding the upper bounds of the (safe) operating frequencies of cryptographic devices, in analyzing the total signal and noise power (which again was covered in this thesis as part of different analyzes detailed next) and may provide specifics to a side-channel attacker to choose points-of-interest based on frequency rather than temporally (as is the prevalent case in side-channel literature today).

In this section, we introduce a mathematical treatment of our introduced methodology; we provide a comprehensive coverage of the various parameters which affect the simulation cost and provide trade-offs which aid both the designer and the evaluator in a cryptographic side-channel scenario.

We further aim at validating all the aspects of the methodology, both mathematically and physically, by (1) defining our target designs and simulation settings, (2) determining the simulation parameters in view of a cost-precision tradeoff, and (3) validating the impact of the low-frequency noise factor by an ac analysis. To be complete and fully guarantee the meaningfulness of our results, we validates the physics of the first-order statistics of the noise present in the supply current of one of the circuits of interest.

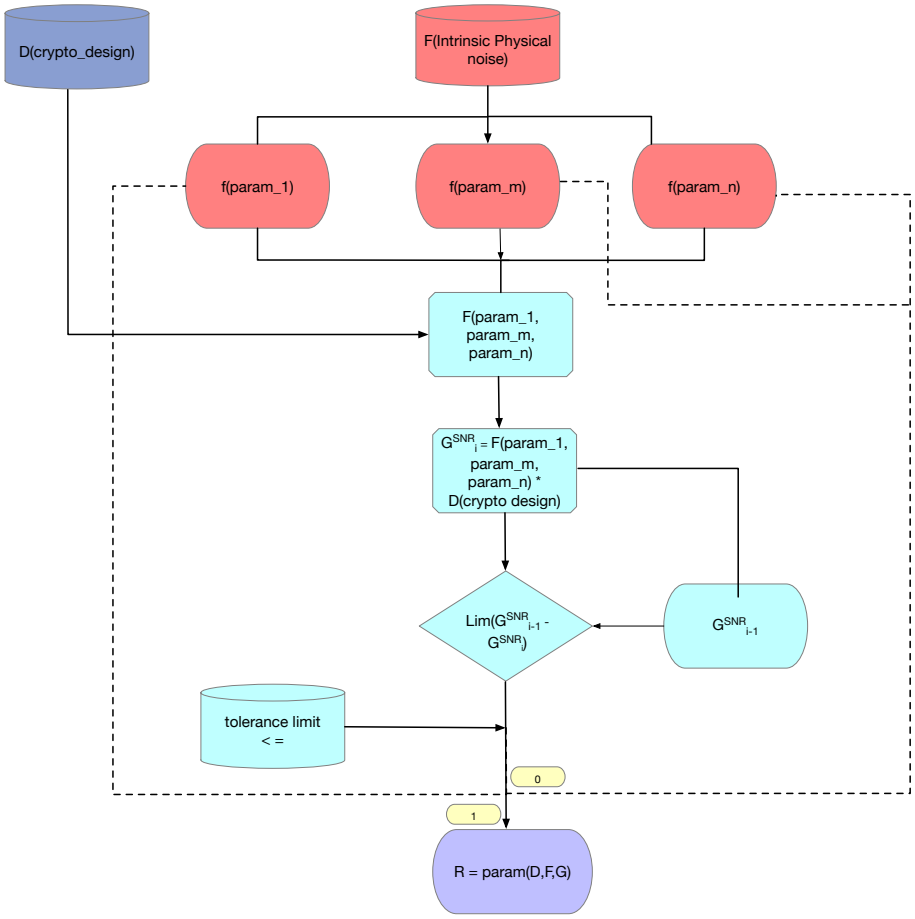


Fig. 4.1: TIMESPAN Methodology for the Evaluation of Intrinsic Physical Noise

A graphical very high-level overview of the proposed methodology can be depicted by Figure 4.1. We subsequently detail the methodology in subsections below. A pseudo-code describing the steps is detailed below (ref. Algorithm 1).

Algorithm 1: TIMESPAN Methodology

Input: f(noise_model), f(crypto_design)
Result: f(nruns), f(fmax)
Require *T, f(nruns), f(fmax)*
Ensure $fmin\_flicker \geq 0$
Data: $\sum_{i=1}^{nruns} := 0$
$fmin\_flicker \Leftarrow 1/T$
$D = f(crypto\_design)$
$fmax \Leftarrow f(fmax)$
$nruns \Leftarrow f(nruns)$
$F = f(nruns, fmax)$
while $i \neq nruns$ **do**
    G_i^{SNR} = f(T, nruns$_i$,fmax$_i$, fmin_flicker, D)
    if $Lim(G_i^{SNR} - G_{i-1}^{SNR}) \leq \eta$ **then**
        while $j \neq 0$ **do**
            $G_{i,j}^{SNR}$ = f(T, nruns$_i$,fmax$_j$, fmin_flicker, D)
            if $Lim(G_{i,j}^{SNR} - G_{i,(j-1)}^{SNR}) \leq \eta$ **then**
                $fmax[i,:] = f(fmax[i,j])$
            else
            end
        end
        $nruns[i] = f(nruns[i])$
        continue
    else
    end
end

An exemplary trace(s) which provide an appreciation of the amount of variation induced in a switching (input) current due to intrinsic physical transient noise is shown in Figure 4.2. Transient noise simulations are performed for a given supply voltage and 100 such runs are recorded and plotted. The clear differences in the (same) supply current account provide an interesting visualization of the induced variations in the switching current, thanks to the IPN.

4.2.1 Proposed Methodology

As mentioned in the preceding chapter, the estimation of the `Signal` values (from the numerator of equation 2.30) was carried out in a *noiseless* setting, i.e., the simulation was in a *noise-free* environment. Although it provided appreciable values for the computation of the signal, and

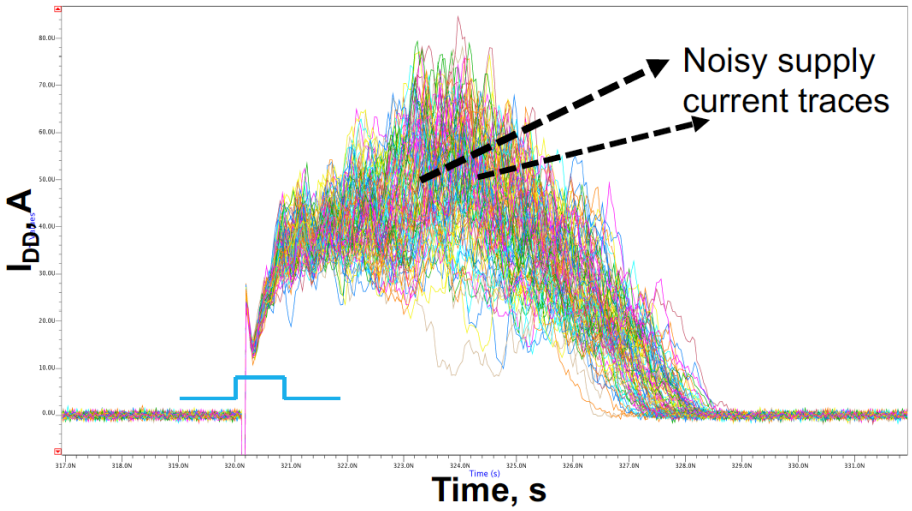


Fig. 4.2: Exemplary traces from a single AES Sbox switching input across 100 transient noisy traces showing the presence of Intrinsic Physical Noise on the current consumption

led to interested observations on the impact of technology scaling for dual-rail and CMOS based devices, an estimation of the noise from the implementation was not possible. The introduced methodology in this chapter provides a broad framework which is able to compute the security evaluation metrics (such as the SNR, MI or PI) for design stage evaluations in a simulational setting. The proposed methodology is able to present the results of the security evaluations from the pre-layout netlist; although the same can be extended to post-layout simulations. However, focus has been limited to pre-layout design stage, as this forms the crux of the design time; any findings as such can be reverted from this stage comparatively easier than reverting from say, a post-layout evaluation.

4.2.2 Physical Noise versus Simulated Noise: The Simulation Problem

In physics, every *intrinsic* noise source has an *infinite bandwidth*. Let us take the white thermal noise as the most trivial but illustrative example:

$$S_{i,\mathrm{th}}(f) = \frac{4kT}{R}. \tag{4.1}$$

PSD such as (4.1) describes how the expected average intrinsic noise power (or simply the average intrinsic noise power if ergodicity[1] is assumed) is distributed in frequency:

$$\sigma_i^2 \equiv \int_0^{+\infty} S_i(f)\,\mathrm{d}f \quad \text{(definition of the PSD)}. \tag{4.2}$$

Plugging (4.1) into (4.2), we quickly observe the following apparent paradox :

$$\sigma_i^2 = \frac{4kT}{R} \cdot \infty = \infty ?$$

If one tries to perform the same calculation using rather a flicker noise PSD, such as (2.14) presented in section 2, he would again obtain a theoretical infinite noise power.

Such a paradox is fortunately solved by experiment: one never measures such an intrinsic noise source. We mean that there is always some *filtering* or *band limitation* effect arising from the measurement equipment on one hand, from the circuit or device under measurement itself. The MOS transistor provides us a great example to illustrate that fact. The drain current noise $i_\mathrm{D}(t)$ is measured as voltage fluctuation $v_\mathrm{o}(t)$ at the output, and what one *observes* is actually the signal

$$v_\mathrm{o}(t) = H\left\{i_\mathrm{D}(t)\right\}, \tag{4.3}$$

where $H\{\cdot\}$ denotes the linear filtering action. The *Wiener-Khinchin theorem* relates the output noise PSD to the input one:

$$S_{v_\mathrm{o}}(f) = \left|H(f)\right|^2 S_{i_\mathrm{D}}(f), \tag{4.4}$$

$H(f)$ being the (deterministic) *transfer function* from $i_\mathrm{D}(t)$ to $v_\mathrm{o}(t)$. Due to the presence of small capacitances, there is always a *low-pass*

[1] for a sufficiently large number of samples from a measurement (or simulation), ergodicity is the asymtptotic mean, i.e., a convergence of the time average to the ensemble average

filtering effect which limits the *actual* noise bandwidth and ensures a *finite* noise power. As an example, let us consider the first-order filter

$$H(f) = \frac{H_0}{1 + jf/f_0}, \tag{4.5}$$

and let us compute the output noise power as

$$\begin{aligned}
\sigma_{v_\mathrm{o}}^2 &= \int_0^{+\infty} S_{v_\mathrm{o}}(f)\,\mathrm{d}f \\
&= \int_0^{+\infty} |H(f)|^2 S_{i_\mathrm{D}}(f)\,\mathrm{d}f \\
&= H_0 \cdot 4kT\left(\frac{2g_m}{3}\right) \cdot f_0 \cdot \int_0^{+\infty} \frac{\mathrm{d}f/f_0}{1 + (f/f_0)^2} \\
&= H_0 \cdot 4kT\left(\frac{2g_m}{3}\right) \cdot f_0 \cdot \frac{\pi}{2}.
\end{aligned} \tag{4.6}$$

We have assumed that the drain current noise consists only in thermal noise of (2.12), for the sake of illustration. It is clear that the noise power computed in (4.6) is now finite.

Although it was shown based on an example, the following statement is general: even if intrinsic noise sources are of an infinite bandwidth, these are *never* probed intrinsically, and the power of the noise measured at the output (v_o) or at the supply (i_DD) is finite because of low-pass filtering effects. From the simulation point of view, it therefore appears useless to generate noise with excessively high-frequency content, as high-frequency components are *not* observed and hence, are not expected to influence much the extracted circuit FoMs such as the SNR, or the MI. In all, it appears very sound to limit the noise to some frequency `FMAX`, so that the corresponding supply noise power is computed as

$$\sigma_{i_\mathrm{DD}}^2 \propto \int_0^{\texttt{FMAX}} S_{i_\mathrm{DD}}(f)\,\mathrm{d}f. \tag{4.7}$$

Regarding the flicker noise, a similar discussion should be made regarding the lower integration bound of (4.7): irrespective of the `FMAX`,

$$\begin{aligned}
\sigma_{i_\mathrm{DD,flicker}}^2 &\propto \int_0^{\texttt{FMAX}} \frac{K_f}{f}(f)\,\mathrm{d}f \\
&\propto K_f \log\left(\frac{\texttt{FMAX}}{0}\right) = \infty?
\end{aligned}$$

diverges, which leads to the introduction of a frequency `FMIN_FLICKER` below which the flicker noise is whitened. The existence of such a cut-off frequency arises from the finite duration T of the simulation (or

measurement). As the noise is a zero-mean quantity, the first spectral component that one can actually estimate is the frequency

$$\texttt{FMIN\_FLICKER} = \frac{1}{T}. \tag{4.8}$$

Processes involving fluctuations slower than $\sim 1 - 10 \cdot T$ are treated as *static* or *time-zero variability* in this frame.

Selecting the proper `FMAX`

Circuits used for cryptographic application quickly become more complex than one single transistor operating in DC condition. However, the previous discussion can be extended as follows. Let

$$\left\{ i_{\mathrm{D}}^{(m)} \right\}_{m=1,2,\ldots N_{\mathrm{sources}}}$$

denote the intrinsic noise sources of all the different transistors making the circuit, and let

$$\left\{ H^{(m)}\left(f,t\right) \right\}_{m=1,2,\ldots N_{\mathrm{sources}}} \tag{4.9}$$

be the associated time-dependent transfer functions to the supply rail. The time dependence comes from the fact that a digital gate operates in *large-signal conditions*, the bias point is continuously changing, and the noise is a non-stationary stochastic process with a time-dependent PSD :

$$S_{i_{\mathrm{DD}}}\left(f,t\right).$$

Invoking again the Wiener-Khinchin theorem and using the superposition principle, we compute

$$S_{i_{\mathrm{DD}}}\left(f,t\right) = \sum_{m=1}^{N_{\mathrm{sources}}} \left| H^{(m)}\left(f\right) \right|^2 S_{i_{\mathrm{D}}^{(m)}}\left(f,t\right). \tag{4.10}$$

We now formally present our *methodology* which is used throughout our work for choosing the simulation parameters of interest and extracting the relevant cryptographic figures-of-merit based on such carefully chosen parameters.

Thus, due the the transistor intrinsic capacitances, (4.10) tells us that the supply noise is the sum of several low-pass filtered white and flicker noise contributions, and that i_{DD} is finite bandwidth limited and has a finite instantaneous power.

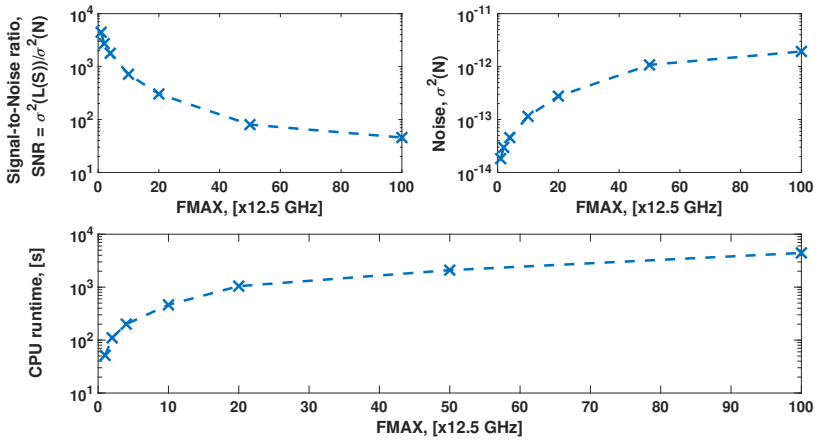


Fig. 4.3: Impact of `FMAX` parameter on the "Cryptographic" *SNR* (computed via ref. Eq 2.30), *Noise* and *CPU runtime* for an XOR gate at V_{DD}=0.9V

As it looks very difficult or at least very tedious and hence impractical to the authors to estimate all the transfer functions (4.9), for a large set of relevant bias points and every time for each circuit architecture, the following empirical methodology was adopted to select `FMAX`. The circuit FoM of interest, such as the SNR, was extracted for a reduced number of `FMAX` values, typically between a few GHz up to the THz. We expected that the noise power *saturates* with a larger `FMAX`, and thereby the SNR *flattens out*. Such a behavior, predicted by (4.10) and the related discussion, is confirmed in Figures 4.3 (top) and 4.3 (top) for a XOR gate. The SNR of reference is unambiguously

$$SNR^* = \lim_{\texttt{FMAX}\to\infty} SNR\,(\texttt{FMAX})\,, \tag{4.11}$$

while the actually computed SNR, for a given finite `FMAX` is always slightly overestimated[2]. However, we can assume that the `FMAX` is large enough once saturation is reached, i.e. for instance

$$\frac{\left|SNR^* - SNR\,(\texttt{FMAX})\right|}{SNR^*} < \varepsilon \sim 10 - 20\%. \tag{4.12}$$

For the purposes of our simulations, we choose an ε tolerance value of $\lessapprox$ 15%. We observe in Figure 4.3 (bottom) that with an increasing `FMAX`, the computation time (here represented by the CPU runtime) increases.

[2] which is especially not a major concern for cryptographic design, as a higher SNR constitutes the worst case for the circuit

Further, since no additional lowering of the SNR or an increase of the noise power is observed with increasing `FMAX`, a lower value of the `FMAX` (chosen such that Equation 4.12) is satisfied. This justifies the usage of lower `FMAX` values to minimize the simulation run-time.

The trade-off is usually very tough, as the CPU time increases drastically with `FMAX`, as illustrated in Figure 4.3 (bottom). For this case study, a `FMAX` of a few hundreds of GHz yields a satisfying estimation of the SNR while keeping the simulation budget reasonable as the total simulation duration remains below one hour. Furthermore, MOSFET-based devices have an intrinsic cut-off frequency in the 100's of GHz range, which require specialized measurement devices such as the VNA[3]; however, for digital measurements, utilizing scopes (e.g., DSOs)[4], are band-limited to a few GHz range only. These considerations are of highest concerns for circuits with rising complexity and number of transistors (such as the Sboxes), as the simulations become more and more computationally intensive.[5]

4.2.3 Simulation cost and Tradeoffs

In this subsection, we investigate the cost of our methodology and quantify the total budget, both in terms of CPU runtime and number of runs (represented by the `nruns` parameter) required to obtain convergence in our metrics which estimate the "crypto" signal and noise. The noise transient simulations are indeed well known to be time-and memory-intensive [27]. Basically, our simulation budget can be stated as

$$N_{traces} \cdot \frac{T}{\mathrm{d}t}, \tag{4.13}$$

where N_{traces} is the number of traces[6], i.e., noise realizations, T is the simulation duration for one trace, and $\mathrm{d}t$ is the time step. These parameters correspond to `nbrun`, `TSTOP`, and `..outstep` of `Eldo .noisetran`

[3]Vector Network Analyzer

[4]Digital Oscilloscopes

[5]In the 4.4.2, after selecting the `.noisetran` parameters as discussed above, we validate that simulated current traces really contain meaningful physical intrinsic noise. For that sake, we briefly examine the first-order statistics of the noise present in the supply current for one of the above mentioned circuits of interest, i.e. the XOR gate. Further, we note that the values in this case are (approx) an order of magnitude higher than those obtained (later) in Figure 4.21; this is attributed to that ref. Figure 4.3 constitutes both the dynamic and static part in the cryptographic noise computations whereas in Figure 4.21 (and Table 4.3), only the static part is considered

[6]N_{traces} is the same as *nruns*[7] below

[78], respectively. The number of time samples for each trace is given by

$$N_s = \frac{T}{\mathrm{d}t}. \tag{4.14}$$

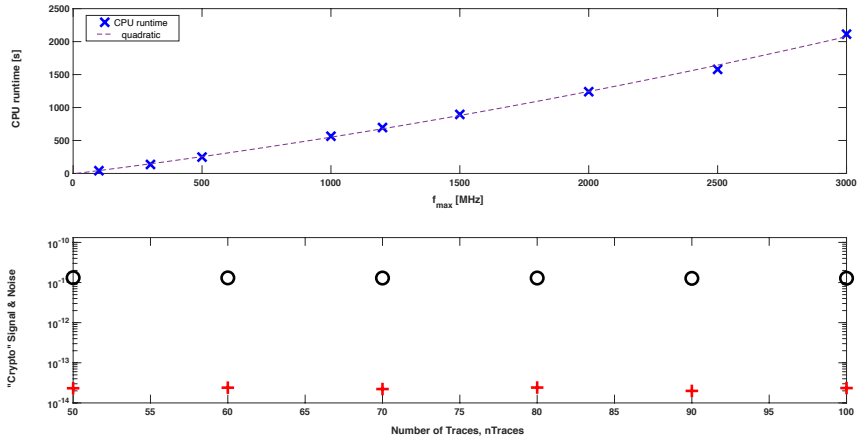


Fig. 4.4: CPU runtime[s] [top] versus `fmax` and computed signal (represented by dots) and noise (represented by plus) [bottom] versus `nruns` for an XOR at V_{DD}=**0.9**V, nruns=100 (for the top half of the figure). FMAX = 1 GHz (for bottom figure). For both, FMIN=1/T, T=10µs

The results from the simulation are illustrated in Figure 4.4, a quasi-quadratic dependency between the `fmax` and the CPU runtime is observed, although at no increased values of computed cryptographic signal and noise parameters. The following parameters have been used: $V_{\mathrm{DD}} = 0.9$V, FMIN $= \frac{1}{T}$, T $= 10\mu$s by varying the `FMAX` and the number of transient noisy runs, `nruns`. This allows us to conclude that given a judicious choice of the `FMAX` and `nruns` choice of parameters, convergence in the computed values for estimating the crypto signal and noise values can be achieved.

4.2.4 AC Analysis

We (very briefly) reiterate noise power computations using a *AC analysis* for a given fixed bias, V_{DD}. We show how the flicker noise is increased by tweaking the model parameters whereas the thermal noise power remains constant. This allows us to investigate the impact of the K_f on the crypto FoMs as discussed in the subsequent sections. From figure 4.5,

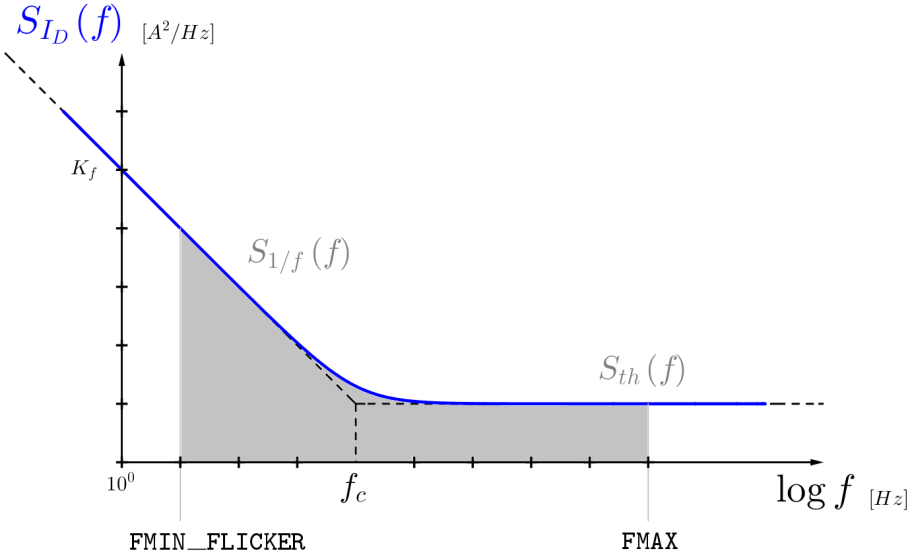


Fig. 4.5: Computation of Noise power using Power Spectral Density

we can compute the total noise power (in this case, the noise variance, σ^2), by integrating the power spectral density $S_{i_{DD}}(f)$ over the desired bandwidth (FMIN[8] and FMAX as defined in the preceding subsections),

$$\sigma^2 = \int_{\texttt{FMIN}_{\texttt{FLICKER}}}^{\texttt{FMAX}} S_{i_{DD}}(f)df \tag{4.15}$$

4.3 Case Study 1: nFET RVT

In this section, as a first example to validate the working of our methodology, we study in detail the intrinsic noise characteristics of a single nFET device. The device corresponds to the fundamental building block of the CMOS28FDSOI technology. We do not differentiate between LVT and RVT devices here; rather we choose the RVT for it being the de-facto standard

[8] we interchangeably use FMIN and $\texttt{FMIN}_{FLICKER}$; they both denote the one and same thing within the scope of this thesis

provided by the technology compared to the LVT, which is provided as a "high performance, higher leakage" variant. Not to miss any effects which would otherwise have been optimized for the workings of the LVT design, we continue our thesis, for all design implementation choices, using the RVT based transistor models. In this section, we show that the intrinsic physical noise simulated using the transient noise analysis, `.noisetran`, matches the same level of noise as that obtained using the classical AC, `.ac` and `.noise` analysis. We also show that how by increasing the `Kf` factor, we can increase the noise by a given factor, in both the analyses. This would tell us the amount of change required to reach a given noise level.

For a better understanding and agreement with the literature [120] available on the subject, we divide our analysis into two parts.

- Firstly, we show that the values obtained using both the analyses (both `AC .noise` and `.noisetran`) are in close agreement with the existing literature.
- Secondly and more importantly, we show that using the effective current [83], for the same voltages, the `.ac` and `.noisetran` produce the same amount of noise. This validates our claim that a similar level of noise can be obtained from the transient noise analysis in noisy transient runs.

4.3.1 Simulated vs Measurement Noise

In this section we show that the noise produced by measurements are in close agreement with the simulated noise (both AC `.noise` and `.noisetran`). The parameters of the experiment are as follows,

- `W` = 270nm, `L` = 37nm. Hence `WL` = 0.01μm^2 vs 0.0025μm^2 for Ghibaudo [120]
- bias (DC) current, I_D = 49.246 nA (corresponding to 47 nA obtained with measurements [70, 71]),
- V_{DS} = 50mV, idem Ghibaudo et al. [120]
- G_{OV} = V_{GS}-V_{th} = -0.1V

From figures 4.6 and 4.7, we see the simulation results are in close agreement with the measurement data [120]. For instance, say at a chosen frequency of `f` = 10kHz, we can clearly see from simulations that $S_{I_D} = 3.5 \times 10^{-24}$ A^2/Hz; from Figure 4.6, although measurements show a huge variation in the values (almost as $\times 10^3$) nevertheless, we

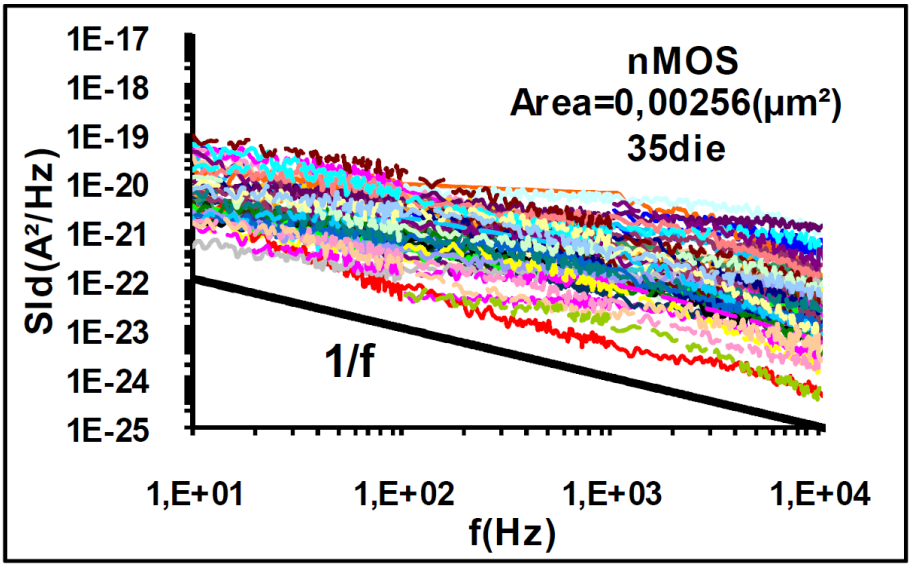


Fig. 4.6: Experimental results of 1/f noise [120] [linear regime]

see for the red trace or the green trace, a conformity with our values is achieved. Additionally, the area difference between the two should also be considered. Since the difference in the respective areas is $\times 4$, we could additionally expect some more noise in the measurements, which is clearly the case. Also not to miss the additional noise which has been induced due to the variability, clearly demonstrated using multiple die measurements, which was not accounted for in our pure transient noise simulations. Since the `.noisetran` would take a computationally very high CPU runtime to simulate traces for 1 Hz frequency, we show the transient noise results from a frequency of 10kHz onwards which are in close agreement with the measurement results (as shown in Figure 4.6).

4.3.2 PSD imbalance using Effective current

In this section, we show that the imbalance in the effective current between the NMOS and PMOS transistors is manifested in the PSD of the two devices. For this, we define the effective current [9] first defined in [83] as

$$I_{eff} = \frac{1}{2} I_D(V_G = V_{DD}/2; V_D = V_{DD}) + I_D(V_G = V_{DD}; V_D = V_{DD}/2) \tag{4.16}$$

[9] which provides a better estimation of the inverter delay and an average of the high and low switching current of the inverter

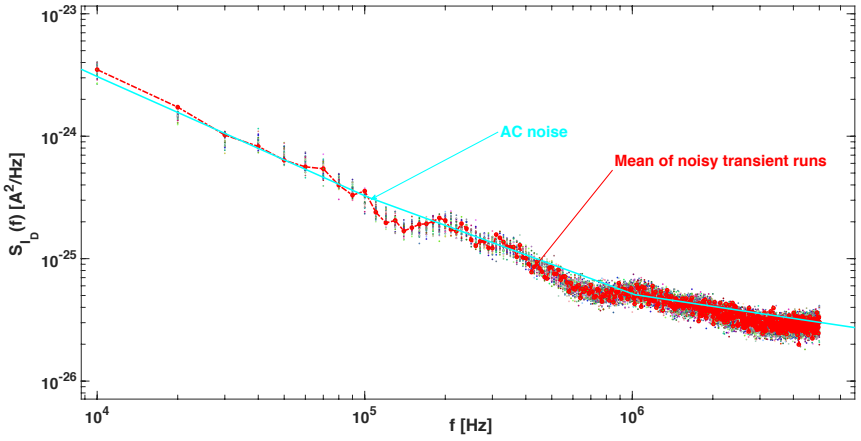


Fig. 4.7: Close matching between [120], AC and Transient Noise analysis [linear regime] corresponding to FMAX=1 MHz for V_D = 50 mV, G_{OV} = V_{GS}-V_{th} = -0.1V of a RVT n-channel MOSFET with W=270nm and L=37nm

Using the simulation parameters described above in section 4.3.1 (V_{GS},V_{DS}, W and L), the effective current for both the PMOS and NMOS devices are calculated and plotted as shown in figure 4.8. The following points are noted from Figure 4.8

1. A clear imbalance between the NMOS and PMOS devices is seen which manifest as difference in the data-dependency observed between the NMOS and PMOS devices. This is of importance to note as useful results are drawn based on this in the subsequent analyses presented.

2. The difference in the data-dependency is clearly exacerbated at lower voltages, especially in the sub-threshold region which decreases on moving to higher voltages; this could suggest the possibility (and which again is a recurrent theme explored in this thesis work), that lower voltages could possibly lead to higher leakages (from a cryptographic perspective).

This mismatch is also manifested in the PSD spectrum of the two devices (obtained from transient noisy simulations).

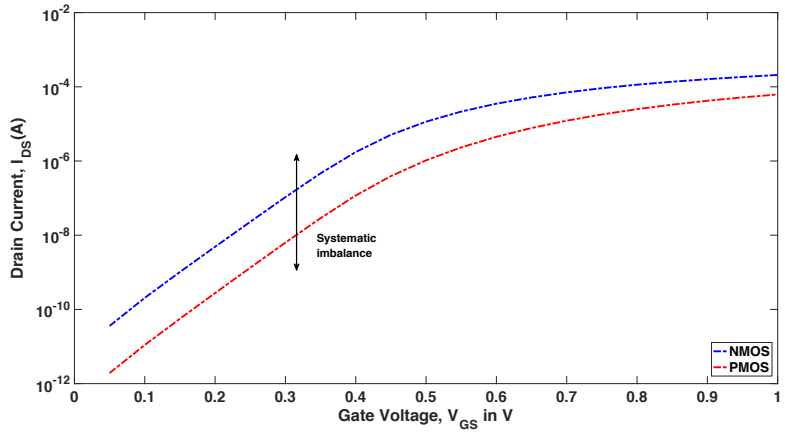


Fig. 4.8: Drain Current imbalance- PMOS and NMOS devices, V_D = 35 mV, W=270nm, L=37nm and FMAX=1 GHz

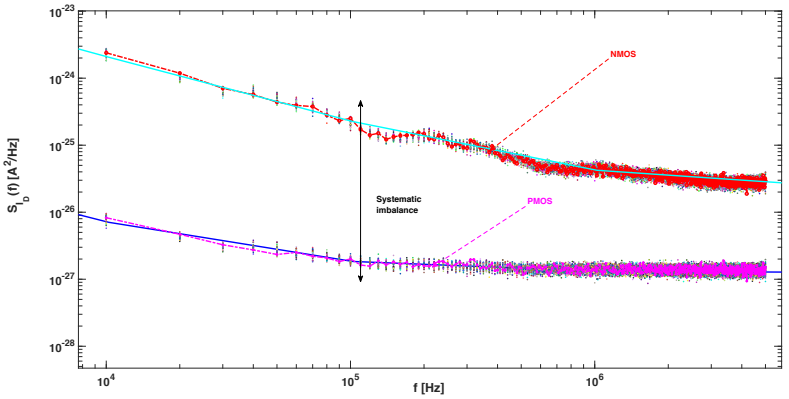


Fig. 4.9: PSD imbalance between the PMOS and NMOS device with (same) dimensions as described in section 4.3.1, bias V_G=0.3V, V_{DS}=50mV, `FMAX`=100 MHz. Solid lines represent the `.AC noise` simulations whereas symbols (or dots) represent the transient noise analyses

Although Figures 4.8 and 4.9 show a clear systematic mismatch between the PMOS and NMOS devices, it may also lead to an (apparent) conclusion that the inherent data-dependency increases while decreasing the supply voltage and hence a higher value of the SNR may be expected

(on account of this apparent mismatch). Fortunately, this anomaly can be explained by accounting for the transistor sizing methodology.

A Discussion on the Drain Current and PSD mismatch and Inherent dependency p-channel MOSFET devices operate on the theory of holes according to semiconductor device physics, whereas n-channel MOSFET operate on the movement of electrons. It is also known from semiconductor theory that holes have reduced mobility (or speed) as compared to the electrons. Consequently, such inherent mismatch in the mobility at the lowest level of physics is manifested as difference in the drain current (at the transistor level) or the difference in the Power Spectral Density (at a mathematical circuit level). Therefore, it becomes imperative that the sizing of the devices are accordingly adjusted to compensate for this mismatch as much as possible. Subsequently, we adjust the sizing (in this case up-size) of the PMOS transistor such that it has the same DC current value as the NMOS transistor (thus effectively equalizing the drain current mismatch and reducing almost (completely)[10] the inherent data-dependency). This allows for a fair comparison of the PSDs, given that we have equalized the current differences.

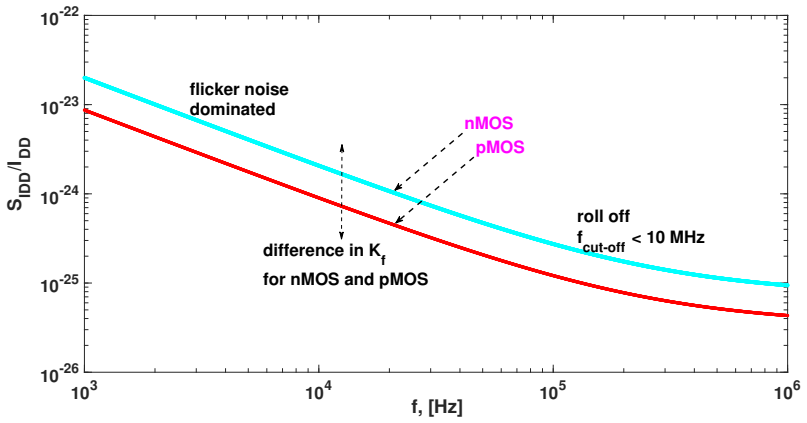


Fig. 4.10: Rectifying the apparent imbalance via `.AC noise` in the PSD (im)-balance between the PMOS and NMOS devices, bias V_G=0.3V, V_D = 35 mV, W=270nm, L=37nm

Figure 4.10 plots the $S_{I_{DD}}/I_{DD}$ obtained via `.AC noise` simulations against the frequency for both PMOS and NMOS device and presents the results of such an evaluation. Although not by a huge factor, but

[10] although we do not claim to have made it zero, but again this is a topic beyond the scope of this work and we leave this for the interested reader as exploratory research

there is a clear difference in the K_f values between the NMOS and PMOS devices. This leads to the following discussions:

1. A clear difference in the K_f between the PMOS and NMOS devices is shown even though the DC current levels are the same. This leads us to a very interesting and relevant conclusion (which is further exploited when we discuss the static region of a 2-bit XOR) namely, that the difference in the output noise (or current) values stems from the difference in the mismatch between the PMOS and NMOS device's K_f and depending upon the number of transistors (and their corresponding g_ms) may affect the overall output function.

2. Secondly, at lower voltages, clearly the cut-off frequency of the transistor(s) rolls-off at a quicker rate than at a corresponding higher voltage, which may increase the overall noise (as the integrated noise is available over a larger bandwidth viz-a-vis at lower voltage). However, at higher voltages, the signal dominates the noise and thus leads to an overall increased SNR.

4.3.3 Impact of `FMAX`

In this section, we show the influence of the `fmax` parameter, the maximum frequency of the noise generating sources, on the cut-off (or roll-off) frequency (at which the noise changes from *orange* to *green*). We keep the same simulation setup as described in the previous section.

The `.NOISE(AC)` simulation in the frequency domain cheaply provides the drain current noise PSD S_{I_D}(f) at any desired fixed DC bias point. The purpose of such simulation is to quickly extract the transistor noise model and thereby to give a PSD of reference. Focusing on the simulation of large-signal nonlinear circuit, it is also instructive to extract this PSD from a real simulation in the time domain, that is a transient noise `.noisetran` analysis. The noise PSD can be estimated using a spectral estimation method performed afterwards in Matlab.

Figure 4.11 compares the reference PSD with different estimates. It aims at illustrating the effect of the `fmax` parameter on the bandwidth of the generated intrinsic transistor noise. We emphasize that all the simulated signals are sampled in time. We have denoted by dt the largest simulation time step, which is also the sampling period of the measured signal id(t). The usual practice for transient noise simulation is to select dt according to `fmax` or conversely. The parameter dt actually determine the largest frequency that we are able to observe [129] [11]

$$f_{obs} = 1/(2.dt) \tag{4.17}$$

[11] the below 2 (paragraphs) discussion on time-step and the observable frequency is

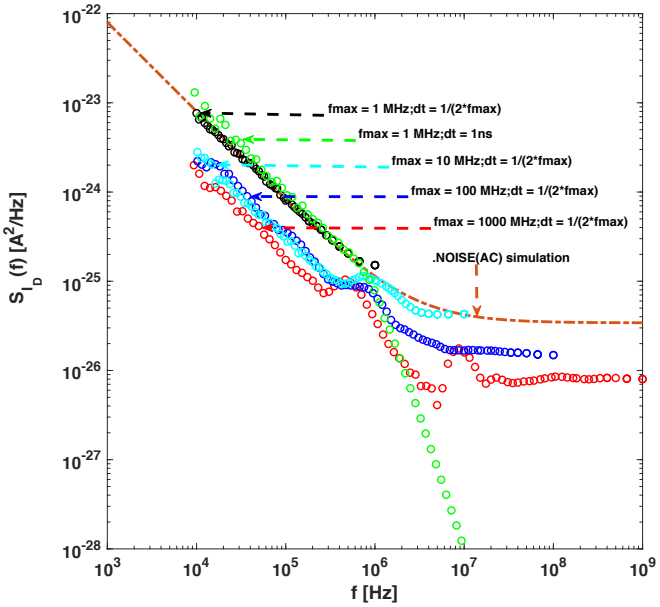


Fig. 4.11: Influence of `fmax` on the PSD for bias V_G=0.3V, V_{DS}=50mV, W=270nm, L=37nm NMOS device, Temperature=25°C, `fmin`=10 kHz across all cases.

Generally speaking, it does not make sense with regard to limited computational resources to simulate frequencies beyond the noise bandwidth `fmax` as those do not impact the circuit behavior. This is why most of the time f_obs matches with `fmax` and why the time step dt is unequivocally related to `fmax` as

$$dt = 1/(2 \cdot \texttt{fmax}) \tag{4.18}$$

However, we may want to observe beyond `fmax`, by selecting a smaller dt such that f_{obs} exceeds `fmax`, in order to assess the quality of the noise generation method [129]. This has been achieved for the `fmax` = 1 MHz case of Figure 4.11; the black dots correspond to the standard circuit simulation framework (*), while the green dots represent the PSD estimated from a signal finely sampled at dt = 1/(1000.`fmax`), which allows us to inspect the behavior of the PSD of the simulated noise. We can observe that the noise power is not strictly confined in the specified [0, `fmax`] frequency band. A part of the noise part is wasted outside the noise

adapted from part of recent collaborative work with Léopold V. Brandt and has been solely added for completeness to make for a coherent and complete reading

bandwidth. This quantitative evaluation of the noise generation methods is beyond the scope of our own work but is expected to be addressed in [129]. The intuitive explanation of this fact is taught by signal processing theory: it is impossible to generate a perfectly, infinitely abruptly band-limited signal of finite duration, by analogy with the impossibility of implementing an ideal low-pass filter. We notice in Figure 4.11 that the roll-off is as steep as $\approx 1/f^3$ in the white noise region.

Finally, we point out the discrepancies between the reference PSD and the PSD's estimated from time-domain traces simulated at larger `fmax` (cyan, blue and red dots). This fact would deserve deeper investigation but can confidently be attributed to signal-processing considerations. Given limited computational time and output file memory size, there is a tough tradeoff between the simulation time step (and the related `fmax`) on one hand, and the number and the duration of the traces on the other hand. Thus, a larger `fmax` implies to work with a reduced number of traces more limited in duration. We know from spectral estimation theory [117] that too short available realizations can lead to significant bias in the subsequent PSD estimates.

We additionally observe that increasing the `fmax` (may) not affect the magnitude of the noise; this is partially true as the PSD is affected, total integrated noise may be not, and that is maybe the reason of the PSD decrease in transient noise mode with `fmax`, the total noise is distributed differently depending on the dt getting shorter and shorter; and since the total bandwidth is increased, the computation of the total noise variance (ref. Equation 4.15) increases. Clearly, the above interpretations resonate well with Figure 4.3, wherein similar convergences were observed when increasing the `fmax` parameter and observing an increase in the noise.

4.3.4 Influence of K_f on Noise, $\mathtt{S}_{I_D}$

The power spectral density, or the $\mathtt{S}_{I_D}$ is a measure of the noise w.r.t the frequency. Very simply, at $f = 1$ Hz, the $\mathtt{S}_{I_D}$ value is known as the K_f. This factor K_f is used by commercial simulators to increase or decrease the amount of noise being generated. In the current 28nm FDSOI technology being used, this factor can be simulated by changing the `nsig_dnoise` parameter. Referring Equation 2.14, we see that an increase in the K_f leads to an increase in the value of the noise.

From figure 4.12, we can make the following observations:

1) With an increase in the value of the `nsig_dnoise`, the magnitude of the noise increases. A x1 increase causes a x10 increase in the S_{I_D} (similar conclusions can also be indeed derived from the `.ac noise` simulation results presented in Table 4.1), whereas an increase of

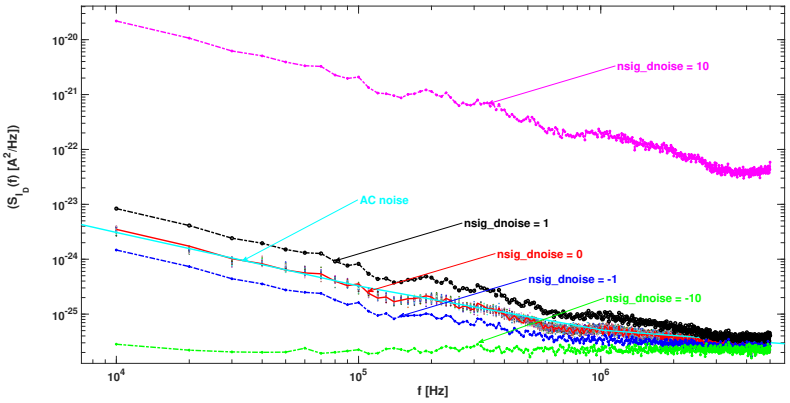


Fig. 4.12: Influence of `nsig_dnoise` parameter on the PSD [in log scale], for a *transient noise* simulations for a n-channel MOSFET RVT device of W=80nm, L=30nm, V_G=0.5V, V_{DS}=35mV, Temperature=25°C

Table 4.1: Impact of K_f parameter on the Noise for a n-channel MOSFET RVT device at V_{DD} = 0.5V,V_G = V_{DD}; V_D = 35mV; W=80nm, L=30nm under *.ac noise* simulations

Flag	Corresponding $K_f = 10^{nsig\_dnoise}$	$S_{I_{DD}}$, 1 Hz [A^2/Hz]	σ^2, [A^2]
nsig_dnoise = 0	1	4.13×10^{-19}	4.76×10^{-18}
nsig_dnoise = 1	10	1.96×10^{-18}	2.267×10^{-17}
nsig_dnoise = 4	10000	2.11×10^{-16}	2.44×10^{-15}
nsig_dnoise = 10	10^{10}	2.45×10^{-12}	2.83×10^{-11}
nsig_dnoise = -1	0.1	8.68×10^{-20}	1.04×10^{-18}
nsig_dnoise = -4	0.0001	8.07×10^{-22}	1.14×10^{-20}
nsig_dnoise = -10	10^{-10}	9.07×10^{-26}	2.105×10^{-21}

x10 in the `nsig_dnoise` increases the S_{I_D} by a factor of almost $\times$10000

2) With lowering of the `nsig_dnoise` value, i.e by indirectly lowering the K_f factor, we see that the decrease is by a factor of $\times$100 for a $\times$10 decrease of `nsig_dnoise` and the noise becomes more white. Therefore it suggests, that decreasing this factor reduces the flicker noise in the system which is validated by Equation 2.14. Again, similar conclusions can also be arrived from the `.ac noise` simulation results presented in Table 4.1.

3) Thanks to the increased noise, we now have a better correlation of the noise values represented by Figures 4.6 and 4.12. Indeed, an increase of $\times$10 in the `nsig_dnoise` corresponds to noise values

which closely matches with the measurements as obtained by [120]. Interestingly, we remark here an interesting conclusion and as point of future research work, namely, that the noise induced due to Monte-Carlo simulations (corresponding to the repeated measurements across different dies) in addition to transient noise simulations could substantially lead to an overall higher noise than repeated measurements on the same die (corresponding to transient noise simulations exclusively without accounting for PVT variations).

It is worth mentioning here that reader may notice a change in the values of the `W` and `L` in Figure 4.12 from the preceding subsection. The discrepancy is by design and is attributed to the choice of minimum values so as to maximize the impact of the flicker noise (ref. Equation 2.14. Subsequent presented results consider the W=80nm, L=30nm for the n-channel MOSFET RVT devices and the correspondingly scaled dimensions for the p-channel MOSFET RVT devices (unless otherwise specified).

4.3.5 Impact of the Low-Frequency Noise factor

We assume a canonical and general expression for the flicker noise (though actual models in `ELDO` are more complex),

$$S_{i_{\mathrm{D,flicker}}}(f) = \frac{K'_f}{f}^{12}. \tag{4.19}$$

From (4.19), the interpretation of K'_f is quite obvious :

$$K'_f = S_{i_{\mathrm{D,flicker}}}(1\mathrm{Hz}). \tag{4.20}$$

For our study of the impact of the transistor intrinsic noise on the SNR and MI, we have tuned, i.e. increased, the noise by increasing the value of such a K_f for our simulations. Such an approach could firstly sound very artificial, but it is supported by Low-Frequency Noise (LFN) measurements from the literature, shown in Figure 7 in [49] for the same technology. Among the samples, many transistors exhibit K_f values more than one decade larger that the one we have artificially introduced. An interesting conclusion that arises from this discussion is the actual possibility to obtain dies within which a large number of transistors achieve a large K_f, as desired to make noisy cryptographic circuit design.

[12] we note here the difference in the flicker noise factor with (2.14); Here and unless specified, the $C_{ox}^2 WL$ term is included in the K_f term

Discussion on the Kf factor We discuss the impact of the increase of K_f factor on a single transistor, the nFET RVT device and a 2-bit XOR gate (in the subsequent section). Our choice of these devices is motivated by the flexibility which such simple designs offer. Additionally, [49] offers an interesting comparison between actual measurement values and our simulated values. As aforementioned, increasing the K_f factor provides a method by which the noise, added to a system intrinsically, can then be used for further security evaluation.

Table 4.2: Impact of K_f parameter on the Noise for a n-channel MOSFET RVT device at $V_{\mathrm{DD}} = 0.9\mathrm{V}$,$V_{\mathrm{G}} = V_{\mathrm{DD}}/2$; $V_{\mathrm{D}} = V_{\mathrm{DD}}/2$; W=80nm, L=30nm

Flag	Corresponding $\mathrm{K}_f = 10^{nsig\_dnoise}$	$\mathrm{S}_{I_{DD}}$, 1 Hz [A^2/Hz]	σ^2, [A^2]
nsig_dnoise = 0	1	3.19×10^{-18}	5.16×10^{-17}
nsig_dnoise = 1	10	1.51×10^{-17}	2.45×10^{-16}
nsig_dnoise = 4	10000	1.63×10^{-15}	2.63×10^{-14}
nsig_dnoise = 10	10^{10}	1.89×10^{-11}	3.05×10^{-10}
nsig_dnoise = -4	0.0001	6.23×10^{-21}	2.73×10^{-19}

Single RVT device and Kf From Table 4.2, we observe an increase of two decades in the noise for a 10^3 increase of the K_f factor. Although such increments, may at first, seem implausible, practical measurements, most notably from, [49], seem to support our hypothesis. We however, do note here, that the increased values are indeed a result of the variation of the `nsig_dnoise` parameter, which was available from the 28nm design PDK. Although the inherent equations (may be) proprietary, we nevertheless emphasize that the first order equations generally considered (such as equation 4.19) do not fully account for the complex, non-linear behavior of IPN in such lower technology nodes. Nevertheless,the values tabulated below, do indeed present a well-expected trend even if considering first order relations.

Although such increased values, as in Table 4.2, may at first glance seem impractical, and we do realize that these values can only be artificially introduced (for instance, in the context of this work and this technology (i.e., 28nm), through the variations of certain parameters (for instance, the `nsig_dnoise` parameter) for the purpose of simulations only), we emphasize that such increased values leads to an interesting test-case evaluation; namely, that if a given certain noise value is desired keeping in mind a target security level (either through the SNR or the PI) they could at least from a simulational perspective be generated and simulated and verified and first hand security FoM can then indeed be

derived. Such pre-fabrication assessment would indeed provide valuable information for a designer to quantify the amount of noise which could further be added from the protocol level.

4.4 Case Study 2 : 2-bit XOR

In this section, building on the validation of the methodology from a single transistor, we extend our analysis to a 2-bit XOR gate that incorporates four NMOS and four PMOS transistors. Understanding the behavior of intrinsic noise in a digital gate would further our understanding of the trends of IPN in a full implementation.

We provide a visual depiction of the noisy current traces from an XOR gate under operation. Although trivial (from a single gate perspective), we nevertheless, provide traces of the computed signal, noise and SNR (ref. Equation 2.30), for the sake of completeness. We further present a statistical characterization of both the dynamic and static regions of a switching XOR gate and finally conclude the section with a discussion on the challenges related to analyzing the noise in the dynamic region.

4.4.1 Visual inspection of traces and correlation with Crypto parameters

In specifying the first-order metrics (such as the SNR used in our case-study), it is often difficult to correlate the cryptographic metric with the corresponding trace (current trace in this case) to have a clear idea of the Point-of-Interest (POI). However, on a closer inspection we see that such a correlation does exist and is useful for visualizing the POI (the maximum SNR in this case) with the "raw" current traces obtained from the experiments. For this, we present here a visual comparison of the "simulated" current traces (in a log scale w.r.t the number of samples), the "crypto" signal and noise, and finally the "crypto" SNR. For the purposes of our experiments, we have the number of samples, n=10000 throughout all our experiments.

Simulation Settings For the purposes of the subsequent experiments, we detail the following conditions; namely, FMIN_FLICKER=1/T, FMAX=1 GHz, nruns = 100, T=10 µs. The dimensions of the transistors have been kept to a minimum to allow for the maximization of the flicker noise (i.e., `W` = 80 nm and `L` = 30 nm for NMOS[13]).

[13]for the PMOS, the device dimensions were correspondingly scaled up to remove any biases from the reduced mobility of PMOS devices

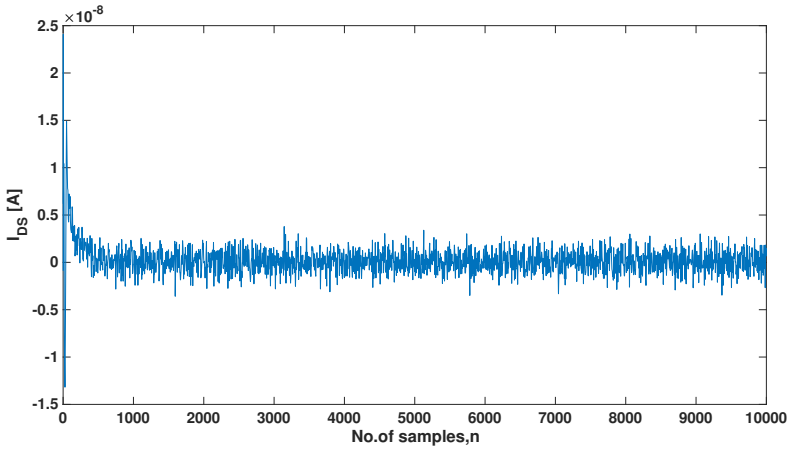


Fig. 4.13: Current trace of a single input transition (00 transition) for an XOR, V_{DD}=0.3V

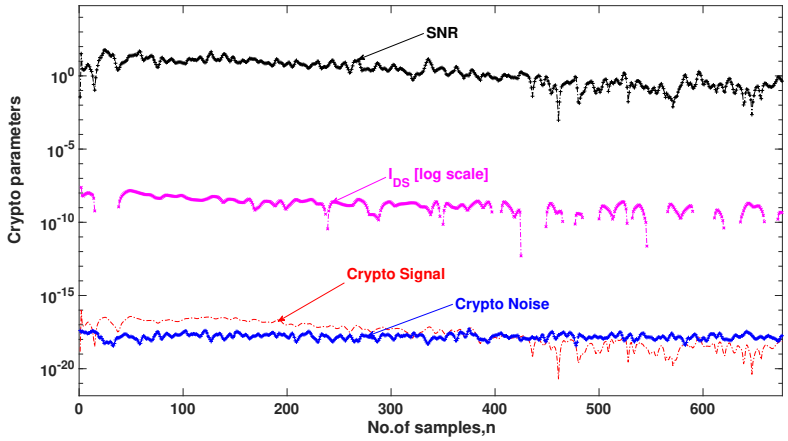


Fig. 4.14: Crypto parameters for an XOR, V_{DD}=0.3V (with zoom on the first 700 samples from Figure 4.13)

From figures 4.13 and 4.16, it is clear that the influence of the noise is more visible at lower supply voltages (0.3V compared to 0.9V). This is also clearly manifested in the corresponding SNR[15] (subsequently validated by Figure 5.7), where the SNR at V_{DD}=0.3V is 59.75 compared

[14]Noisier here does not necessarily mean more noise or increased K_f; it simply indicates that the impact of the noise on the complete waveform is now considered compared to the limited (zoomed) region as previously shown in Figure 4.14

[15]computed throughout the thesis using Equation 2.30

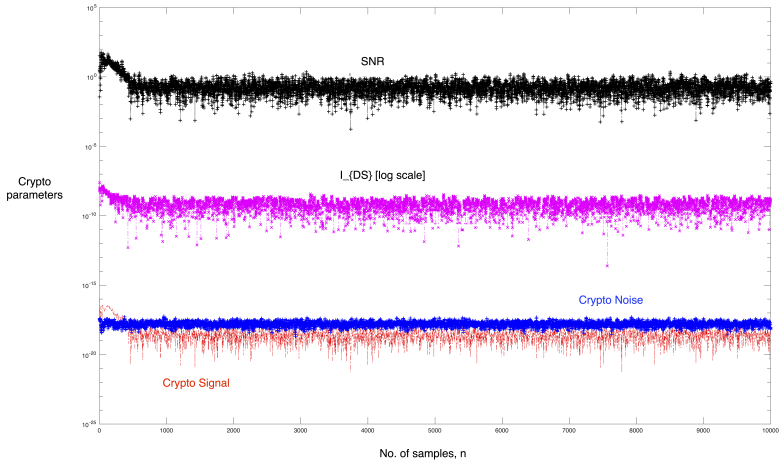


Fig. 4.15: Crypto parameters for an XOR, V_{DD}=0.3V, "Noisier"[14]version.

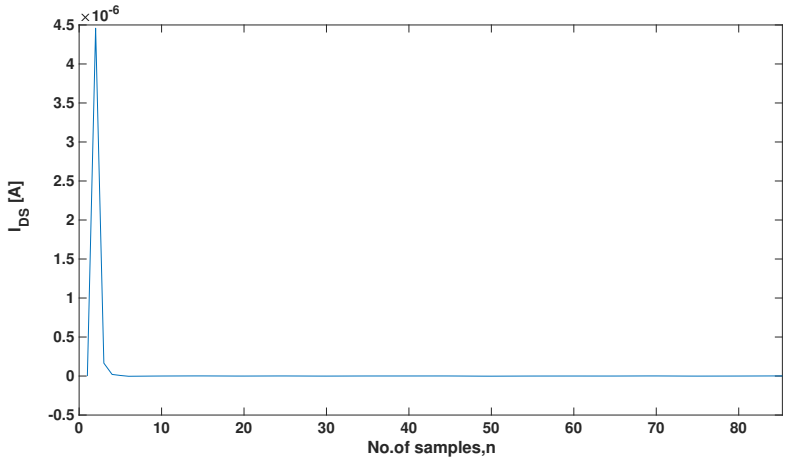


Fig. 4.16: Current trace (noiseless) of a single input transition for an XOR, V_{DD}=0.9V

to the SNR at V_{DD}=0.9V which is 3.46×10^4, a change of $\times$500. The SNR values are computed using Equation 2.30 using traces obtained via multiple runs [16]. More interestingly, the position of the maximum SNR (taking the 0.3V in case, Figure 4.14), closely matches the current trace at the similar position. A "more noisier" (hence a more complete) version

[16]multiple runs are not shown here for brevity; only a single representative case is shown for simplicity

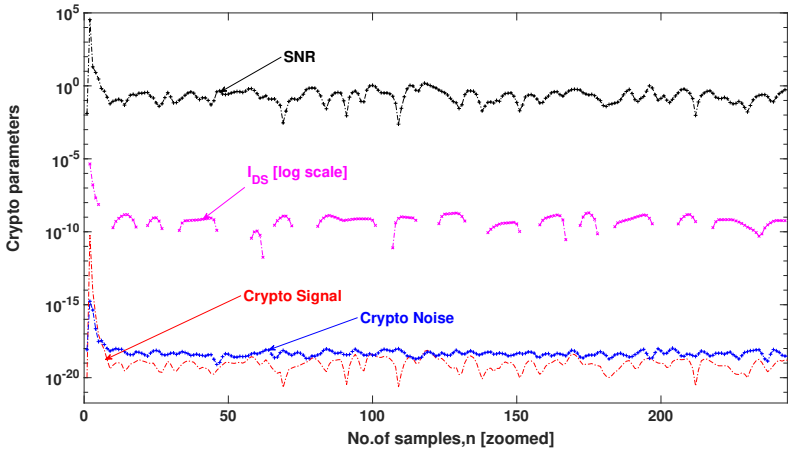


Fig. 4.17: Crypto parameters for an XOR, V_{DD}=0.9V

is shown in Figure 4.15 in which there is a clear correlation between the signal, SNR and the current trace values. Comparing this to the XOR gate for a V_{DD}=0.9V, Figures 4.16 and 4.17, we see there is much less "noise" compared to signal. This is not true considering the absolute noise amplitude. This is simply due to the lesser influence of the noise in this case as also apparent by the differences in the respective SNR values. Of course, the noise at 0.9 V is higher in magnitude compared to 0.3V for the same XOR gate, explained by Equations 2.12, 2.13 and 2.14.

Additionally, the POI, although varies with variation in the supply voltage, does not affect the overall signal and noise calculations. The only difference is observed in the position of the POI, which could be possibly attributed to the increased delay, for e.g., at lower voltages compared to higher voltages. Since we do not average out the results (which is traditionally done in side-channel analysis to smoothen out the noise), we do not lose any information and as such capture the complete information (both in terms of signal and noise). Furthermore, during static analysis, where the SNR could fall below 1, essentially, each point in the trace can be regarded as the POI, thanks to the time-invariant property (as explained in the subsequent section).

4.4.2 Statistical properties of the supply current noise of the XOR gate

In this section, we briefly examine the *first-order statistics* of the noise present in the supply current for one of the above mentioned circuits of interest, i.e., the XOR gate.

Figure 4.18 contains noise traces for the set of parameters $N_{traces} = 40$, T = 800ns and dt = 40ps for a total budget[17] of $\sim 8 \times 10^5$ for a supply $V_{\mathrm{DD}} = 0.9$V. FMIN = 1/T and FMAX = (1/2)$*$dt are chosen as per the described methodology. The four peaks in Figure 4.18 and Figure 4.19 (top part) correspond to the four switching ("dynamic") inputs, thanks to the data applied at the inputs of the XOR-gate. They are in the order 00, 01, 10 and 11 respectively; whereas the longer and non-active regions represent the "static" regions of activity.

Statistical characterization of the static region

In this subsection, we analyze the static region of a digital switching 2-bit XOR gate using `.transient noise` simulations.

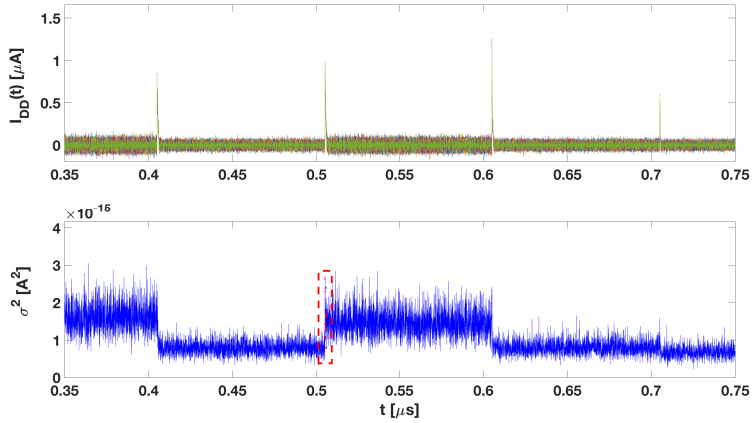


Fig. 4.18: Idd(t) trace for XOR with 4 input transitions; on the dynamic and static regions (top) and associated variance plot based on all the 40 traces (bottom). Input voltage supply = 0.9V

We further include traces for supply voltage V_{DD}=0.3V (Figure 4.19) to provide a comprehensive pedagogical representation for both low and high supply voltages. Qualitatively, similarities are clearly observed at a first glance between Figures 4.19 and 4.18.

On taking a closer look at Figure 4.18, a difference in the values of different static regions are observed; i.e., different static regions corresponding to input transitions of 00, 01, 10 and 11 are observed with different current levels. Since these involve transient noise simulations

[17] defined by

$$N_{traces} \cdot \frac{T}{dt} \tag{4.21}$$

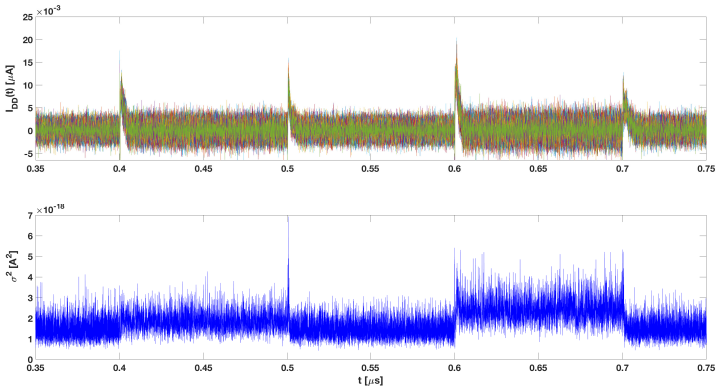


Fig. 4.19: Idd(t) trace for XOR with 4 input transitions; dynamic and static regions (top) and associated variance plot based on all the 40 traces (bottom). Input voltage supply = 0.3V

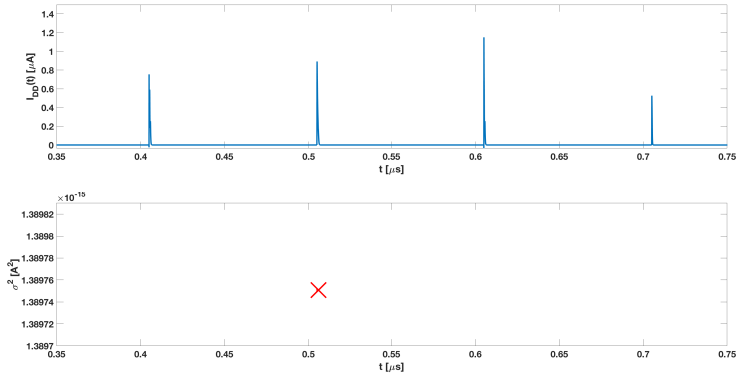


Fig. 4.20: Deterministic transient-noise(less) simulation showing the XOR, for $V_{\mathrm{DD}} = 0.9$V, transitions (top) and the computed signal (below). Since there is only one available trace, hence only the signal (represented by the cross)

across multiple runs, we represent their average values in Table 4.3. Table 4.3 represents the different average current levels for the respective static regions for all the corresponding four input transitions.

Interestingly, the "static" regions are not uniform (or rather the same) across all the regions of longer, non-activity period. This is attributed to the difference in the switching of the corresponding PMOS and NMOS devices. We briefly revisit the equation and expand it in relation to the static power in presence of IPN from section 4.4.2, we obtain below,

$$\left(\sum_{1}^{N_T} I_{leak}(t) V_{\mathrm{DD}} \circ \int_{0}^{fmax} \frac{K_f}{C_{ox}^2 WL} \frac{(g_m^2)(f)}{f} df \right) \tag{4.22}$$

$$V_{\mathrm{DD}} \frac{K_f}{C_{ox}^2 WL} \left(\sum_{1}^{N_T} I_{leak}(t) \circ \int_{0}^{fmax} \frac{g_m(f)^2}{f} df \right) \tag{4.23}$$

From Equations 4.22 and 4.23, we see a non-linear dependence, thanks to the contribution of the different g_m (due to individual transistors). This difference in the corresponding g_m's is attributed to the difference in the static levels of the different regions, as explained below (since different PMOS and NMOS in different configurations are connected).

Table 4.3: Variance of different `static` regions from Figure 4.18 for a 2-bit XOR at $V_{\mathrm{DD}} = 0.9$V from `transient noise` simulations

Region	Mean [nA]	Variance [fA^2]
region_00	0.2535	0.836150
region_01	0.3329	1.463990
region_10	0.2498	0.681278
region_11	0.2382	0.798767

The difference in the noise levels are due to the different "input configurations", i.e., different transistors are turned “on”/“off” differently, so the summed up (or rather the filtered) noise (thanks to the different transfer functions at each different configurations (hence different g_m of transistors coupled with the correlations and interactions of the different capacitances, including with the supply) is different. This systematic analysis of the noise power provides a qualitative appreciation of the different noise levels (in terms of the static voltage/power/current)[18].

Interestingly, from Figure 4.20 (comprising of only the signal), we see a difference in the values with respect to Figure 4.18 (comprising the noise also). Clearly, the effect of transient noise is visible, both in terms of the difference in the quantitative values and a qualitative visual representation of the traces themselves.

Figure 4.21 corresponds to a zoom of the static regions for one of the data of interest (in this case, the "01" inputs). The noise power computed

[18] a quantitative and rigorous exercise is left as an open question for further research; although from a cryptographic perspective, may not be (very) relevant. Cryptographically speaking, the interest in pursuing an attack such as the SCA is motivated by the difference in the *empirical* values of two sets of data, for instance, rather than in the *absolute* values

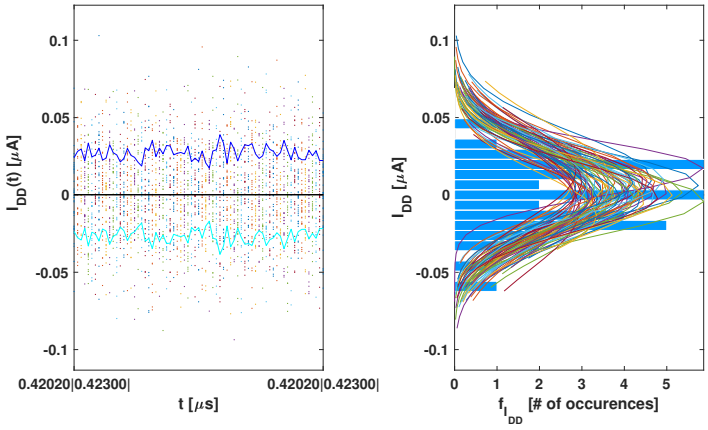


Fig. 4.21: Histogram construction for time t = 0.4202μs - 0.4230μs within the static region of Fig. 4.18 (top). Extracted Gaussian distribution is also shown (right).

from the sigma of the mean Gaussian represented in Figure 4.21 is 1.631×10^{-15} A^2 which is in close agreement with the previously obtained value from Table 4.3

The static region indicated in the current vs. time plots in Fig. 4.18 (top) is useful to get insight on the noise behavior within the circuit and how it affects the supply current. Input voltages of the gate are fixed, and so are all the averages of the branch currents and node voltages within the circuit (since there are noise fluctuations). Hence, the supply current noise is treated as a *wide-sense stationary* stochastic process, for which a complete definition can be found in [93]. Especially, the probability density function (PDF) does not depend on t, that is:

$$f\left(i_{DD}\left(t\right)\right) = f\left(i_{DD}\right), \tag{4.24}$$

and the variance is also independent of t:

$$\sigma\left(t\right) = \sigma. \tag{4.25}$$

As a consequence, every sample $i_{DD}\left(t^*\right)$ at any time sample t^* of every trace is understood as a realization of one single random variable i_{DD}. Figure 4.21 demonstrates that noise in the static region follows a stationary Gaussian distribution.

Correlation of the Static (Transient) Noise values with the AC simulation noise In this discussion, we attempt to correlate the

values obtained via transient noise simulations with ac noise simulations. We elaborate the following conditions for the following analysis:

1. Since the transient noise results mentioned in the preceding discussions are simulated at a supply voltage of $V_{\mathrm{DD}} = 0.9\mathrm{V}$, we retain the same voltage for the subsequent `.ac noise` simulations.

2. Secondly, and more importantly, we note that the `FMAX` frequency in the preceding transient noise simulations is actually 12.5 GHz. This entails that now, the integrated noise variance has to be computed upto a higher value. For exactly this purpose, we re-compute the integrated noise variance of a n-channel MOSFET as well to avoid any discrepancy in results.

3. Finally, we keep the same input conditions as in the preceding discussions, i.e., the inputs to the 2-bit XOR gate are 01 (i.e., a constant fixed bias) of $V_{\mathrm{G}} = V_{\mathrm{DD}}$ ("HIGH" or "1") and $V_{\mathrm{D}} = 35\mathrm{mV}$ ("LOW" or "0").

To emphasize, the values obtained in Table 4.3 have been obtained after `transient noise` simulations. To validate our methodology and to show, that indeed the (static) noise values obtained via transient simulations can be traced back to the .AC noise simulations. Indeed, the values computed in Table 4.4 are in close agreement (of the same order of magnitude) with the values obtained in Table 4.3 and through noisy distributions extractions.

Table 4.4: Re-computed Integrated Noise variance of the "01" `static` region for a 2-bit XOR at $V_{\mathrm{DD}} = 0.9\mathrm{V}$ from `.ac noise` simulations

Input Condition	$S_{I_{DD}}$, 1 Hz [A^2/Hz]	Integrated Noise Variance [A^2]
01	9.83×10^{-17}	4.62×10^{-15}

Consolidating the Noise from an RVT to an XOR In the preceding discussions, we have presented the noise value, specifically the $S_{I_{DD}}$, (in 1 Hz [A^2/Hz]) and the integrated noise, $(\sigma_{I_{\mathrm{DD}}})^2$, of an individual n-channel MOSFET and a 2-bit XOR gate separately. In this discussion, we attempt to consolidate the overall integrated noise and show how a convergence can indeed be obtained. For this analysis, we make use of Equations 4.15 and 4.23.

We recomputed the integrated noise variance of NMOS at $V_{\mathrm{DD}} = 0.9\mathrm{V}$; $V_{\mathrm{G}} = V_{\mathrm{DD}}$ and $V_{\mathrm{D}} = 35\mathrm{mV}$ from `.ac noise` simulations corresponding

to an FMAX=12.5 GHz with $S_{I_{DD}}$, 1 Hz [A^2/Hz] = 1.05×10^{-18} and $(\sigma_{I_{\rm DD}}^{RVT})^2 = 1.12 \times 10^{-15}$ A^2.

We note that the values, as obtained from simulations for $(\sigma_{I_{\rm DD}}^{XOR})^2$ = 4.62×10^{-15} A^2[19]. Recapitulating Figure 4.5, we recollect that FMAX = 12.5GHz, $V_{\rm DD} = 0.9$V for a constant K_f. We also assume that since we are operating in the static region

$$(\sigma_{I_{\rm DD}}^{XOR})^2 = N_T \times (\sigma_{I_{\rm DD}}^{RVT})^2 = 4 \times 1.12 \times 10^{-15} = 4.48 \times 10^{-15} \quad (4.26)$$

where N_T corresponds to the number of switching transistors (in case of `.transient noise` simulations) or "ON" transistors[20] in case of `.ac noise` simulations, which is in excellent agreement with the values obtained from the `.ac noise simulations` of a 2-bit XOR under the same operating conditions as tabulated in Table 4.4. The slightly higher values obtained via `.ac noise` simulations compared to the `transient noise` simulations (although both are of the same order of magnitude) can be attributed to the filtering of the noise values by the output capacitances in case of transient noisy simulations whereas such filtering setups are not encountered in pure `.ac noise simulations`. Similar conclusions are also reached for different voltages and different input conditions, although not shown here for brevity.

Clarification on the components of the dynamic region power

The power consumption due to the dynamic component of a switching trace was briefly introduced in section 2.2. We revisit this in the context of a 2-bit XOR. To reiterate briefly, for a system with N given inputs, the total switching power can be represented as,

$$P_{switching} = \frac{1}{T}\int_0^T V_{\rm DD} I_{switching}^k(t)dt = V_{\rm DD}^2 f_{clk} C_{load} \alpha_F \quad (4.27)$$

where k represents the k_{th} input switching at time t, f_{clk} the frequency of operation of the circuit and C_{load} the total load capacitance, and α_F the switching activity of the circuit represented as the ratio of the probability of the number of switching inputs to the total number of inputs, i.e.,

$$\alpha_F = \frac{\sum_{k=1}^{N} pr_{0\to1}^k}{N} \quad (4.28)$$

$$\sum_{k=1}^{N_T} \alpha_F \left(V_{\rm DD}^2 f_{clk} \Big(\sum_{1}^{N_T} C_{load}(t) \Big) \right) \circ \int_0^{fmax} \frac{K_f}{C_{ox}^2 WL} \frac{\sum_1^{\alpha_F} (g_m^2)(f)}{f} df \Bigg) \quad (4.29)$$

[19] values obtained from ref. Table 4.4

[20] i.e., devices on which a fixed bias (either "0" or "1") has been applied

$$= \alpha_F V_{\mathrm{DD}}^2 f_{clk} \frac{K_f}{C_{ox}^2 W L} \left(\sum_1^{N_T} C_{load}(t) \circ \int_0^{fmax} \frac{\sum_1^{N_T} g_m(f)^2}{f} df \right) \quad (4.30)$$

Equations 4.29 and 4.30, seemingly complex, intuitively capture the characteristics of the dynamic region of the switching power in presence of IPN. Although a formal quantification of the equation is a non-trivial task (and outside the scope of this thesis), we nevertheless attempt to provide an appreciation of the various factors which contribute to an important role. Namely,

1. The parameters marked in blue indicate the parameters directly utilized in this thesis and explained. On account of their behavior being mainly multiplicative in nature, they are expected to increase the total power (or the contributing total variance power) in a given scope.

2. The time (and frequency) varying parameters, related to the $C_{load}(t)$ and $g_m(f)$ respectively, correlate to the non-linearities which contribute to the highly non-linear behavior (as described subsequently). Their summation and corresponding mixture (represented by the $\circ$[21] operation) creates a highly non-linear system, which provides the composite nature of the IPN.

Further, looking at one of the dynamic peaks corresponding to a switching peak, for instance the 01 transition around time instant 0.5 μs in Figure 4.18, we see the maximum value is in agreement with the noiseless case transient simulation as shown in Figure 4.20.

In order to accurately capture the dynamic region behavior, we plot the time-varying histograms of the very narrow dynamic region enclosed by a red rectangle in figure 4.18(bottom). The supply current noise now is a non-stationary stochastic process. Its distribution is explicitly time-dependent:

$$f\left(i_{DD}\left(t\right)\right) = f\left(i_{DD}, t\right), \quad (4.31)$$

as well as the variance $\sigma\left(t\right)$. Since each point in the dynamic region is non-stationary, we observe the *time-varying* histograms for each sample to extract the mean and the variance and show the envelope of the $\pm 1\sigma$ over the mean trace as shown in figure 4.22

Our main goal is to extract the variance from the histograms. This is achieved by performing a nonlinear fitting of the histogram. Using

[21] Intuitively, the $\circ$ could well be represented by a Taylor's series expansions with coefficients of varying units

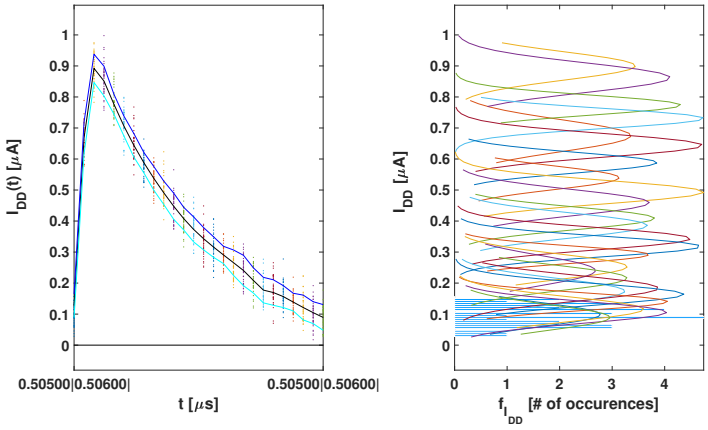


Fig. 4.22: Histogram construction for time t = 0.505μs - 0.506μs within the dynamic region marked red in Fig. 4.18(bottom). Extracted Gaussian distribution mixture is also shown(right).

the extracted σ values from the distributions above and comparing them with the mathematical "cryptographic" noise values calculated by (2.30), we obtain a good matching between the data, thus validating the fact that the noise present in the mathematical calculations can indeed be traced back to the results of the transient noise analyses including the MOSFET noise sources.

Ensemble and Time Average(s) and its relevance to computation of the total Noise trace The time average for a given system allows us to characterize the noise of a given system. On account of its stochastic property and the fact that the dynamic region of a switching XOR (or any other digital gate) is a transient varying signal (i.e., highly non-stationary), the averaged quantity over multiple temporal points allows for estimating the total noise of a given system (a switching XOR in this case), considering both the dynamic and *static* regions, rather than as individual systems spread spatially (referred to as the ensemble average). Therefore, the time average allows for the estimation of the noise power of the complete trace, across all the transient points whereas the ensemble average allows for the estimation of the noise across all the traces for a given temporal point. We compute the power in the dynamic regions across a given temporal point as the $\sigma^2_{dynamic}$ whereas the (total) noise computed from the time average as the σ^2_{IPN}.

Furthermore, since the `.ac noise` simulations provide us a convenient method for estimating the noise in a given static region, we can (also)

indirectly estimate the noise power in the dynamic region by simply subtracting from the total noise power (as obtained above via the time averaging method), the static noise (.ac) power.

We define the ensemble average for a given system which produces a current sample Id(t) belonging to the sample space IDD across total time t (in this case the length or the total simulation time t) as

$$var\_E(t) = \mathrm{E}[(Id(t) - IDD)^2] \tag{4.32}$$

for every t. Equation 4.32 provides us the variance of the current sample at any given instant t. We know from Equation 2.30, that the noise inherently is the mean of the variance of the signal. Hence, we formulate

$$\mathrm{E}[var\_E(t)] = \sigma^2_{dynamic,\mathsf{IPN}} \tag{4.33}$$

Equation 4.33 estimates the total (switching) noise of a given digital system and correlates with the noise obtained from cryptographic simulations back to the physics-based calculations.

The time average is classically defined as

$$var\_T = \frac{1}{T}\sum(Id - IDD).^2 \tag{4.34}$$

(for each trace of length T). Equation 4.34 provides us the variance of the current sample at across all time samples of a given trace (i.e., both dynamic and static regions). Again, from Equation 2.30, that the noise inherently is the mean of the variance of the signal, we formulate

$$\mathrm{E}[var\_T] = \sigma^2_{total,\mathsf{IPN}} \tag{4.35}$$

. Equation 4.35 estimates the total (switching + static) noise of a given digital switching system and correlates with the noise obtained from cryptographic simulations back to the physics-based calculations.

Related Discussion on the consolidation of the Noise power using the Ensemble and Time average We consolidate here the noise power integrated in the same bandwidth in a similar static bias. For this discussion, we refer the reader to ref. Table 4.3 and Table 4.5. We only provide for the concerned noise power values for the case of interest, i.e., for supply voltage $V_{\mathrm{DD}} = 0.9\mathrm{V}$. Nevertheless, for completeness, we point out that similar results have been obtained for the remainder of the supply voltages as well.

For a 2-bit XOR, at $V_{\mathrm{DD}} = 0.9\mathrm{V}$, with dynamic switching inputs (i.e., with (equiprobable) 00, 01, 10 and 11 transitions) given by ref.

Figure 4.21. Since for each static region transition (i.e., 00, 01, 10 or 11), we have the same number of transistors ON (and OFF), but varying g_m, we average the static power to maintain coherency, hence

$$\sigma^{avg}_{\mathsf{IPN}}[\mathrm{A}^2] = \left(K'_f \sum_{1}^{N_T} \int_0^{fmax} \frac{g_m(f)^2}{f} df \right) \tag{4.36}$$

We first compute the power in the static noise region using the traces obtained via transient noise simulations of ref. Figure 4.21. For this, we compute the averaged noise power from the four static regions corresponding to the (equiprobable) 00, 01, 10 and 11 transitions (from Table 4.3 corresponding to $V_{\mathrm{DD}} = 0.9\mathrm{V}$) and obtain

$$\sigma^2_{static} = 9.4 \times 10^{-16}[\mathrm{A}^2] \tag{4.37}$$

Next, we compute the noise power due to the dynamic switching region as given by ref. Equation 4.33. From this we obtain,

$$\sigma^2_{data} = 9.9 \times 10^{-16}[\mathrm{A}^2], \tag{4.38}$$

So, summing ref. Equations 4.37 and 4.38, we obtain $\sigma^2_{total} = 1.94 \times 10^{-15}$ [A^2]. Next, from ref. Equation 4.34 and 4.35, we obtain the total noise (directly from a transient current trace) as

$$\sigma^2_{total} = 2.08 \times 10^{-15}[\mathrm{A}^2] \tag{4.39}$$

Furthermore, the total noise computed using a cryptographic perspective, i.e., from ref. Equation 2.30, provides us a total noise value of $\sigma^2_{total} = 1.98 \times 10^{-15}\mathrm{A}^2$.

Applying Equation 4.36 and summing up Equations 4.38 and 4.37, we see a match with 4.39. Further, the values (corresponding to the static regions computations) obtained from such calculations are consistent with the integrated (static) noise power variances (across different frequencies) obtained in Tables 4.4 and Table 4.5.

Indeed, we see a close agreement (for the same order of magnitude) between the values obtained directly from the noise power computed via time averaged methods and those computed as the sum of the (averaged) dynamic and static components and all these values can then be re-traced back to those obtained via SNR computations from a pure cryptographic perspective.

4.4.3 2-bit XOR - Kf and Supply Voltage

The impact of the supply voltage change is analyzed on the noise for a given K_f value for the nFET RVT and 2-bit XOR devices in Table 4.5. The provided $\mathrm{S}_{I_{DD}}$ values are computed through a `.AC` analysis, similar to as presented in section 4.3.

A note on the choice of `VG` and `VD` Essentially, what does `.AC NOISE` mean for a digital circuit? Basically, it is the sum of individual transistor noise source contributions at the output node, however at a fixed DC bias point (either static, retention mode gate, or a presumably representative point of the whole dynamic). So then which point do we select? A difficult choice indeed.

However, $V_{\mathrm{G}} = V_{\mathrm{D}} = V_{\mathrm{DD}}/2$ would be easier to justify because we are both at the middle of the DC transfer curve and in an average dynamic point. To illustrate a better visual representation of the applied voltage bias points in a 2-bit XOR, we refer the reader to Figure 4.23 below.

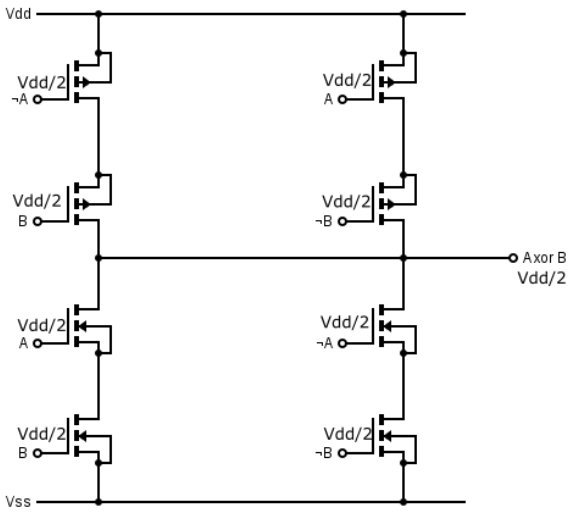


Fig. 4.23: A simple 2-bit XOR under .ac noise simulation conditions with $V_{\mathrm{G}} = V_{\mathrm{D}} = V_{\mathrm{DD}}/2$

Furthermore, for coherency, we retain the same input data conditions as from preceding discussions, i.e., `"01"` for the subsequent below analyses (for the 2-bit XOR case).

From Table 4.5, we analyze the impact of the supply voltage variation on the I_{DD} noise for the same nFET RVT and 2-bit XOR devices. We make the following observations:

- Lower voltages expectedly lead to lower output noise values. This is of importance to note that when operating in the sub-threshold region, although efficient for low power, does not necessarily translate to a secure design, considering the impact of IPN. However,

Table 4.5: Impact of Supply Voltage variation for a given K_f on the computed Noise for an n-channel MOSFET and a 2-bit XOR under `.ac noise` simulations; $V_{\mathrm{G}} = V_{\mathrm{DD}}/2$; $V_{\mathrm{D}} = V_{\mathrm{DD}}/2$; W=80nm, L=30nm NMOS device, Temperature=25°C

Voltage	$\mathrm{S}_{I_{DD}}$, 1 Hz [A^2/Hz]	$(\sigma_{\mathrm{I_{DD}}}^{RVT})^2$	$\mathrm{S}_{I_{DD}}$, 1 Hz [A^2/Hz]	$(\sigma_{\mathrm{I_{DD}}}^{XOR})^2$
	RVT		XOR	
0.3V	3.75×10^{-24}	2.12×10^{-22}	8.31×10^{-21}	5.35×10^{-19}
0.4V	7.68×10^{-23}	1.91×10^{-21}	5.44×10^{-19}	1.07×10^{-17}
0.5V	1.45×10^{-21}	2.63×10^{-20}	4.89×10^{-18}	8.16×10^{-17}
0.6V	2.16×10^{-20}	3.60×10^{-19}	1.69×10^{-17}	2.75×10^{-16}
0.7V	1.99×10^{-19}	3.23×10^{-18}	3.52×10^{-17}	5.70×10^{-16}
0.8V	9.99×10^{-19}	1.62×10^{-17}	5.52×10^{-17}	8.92×10^{-16}
0.9V	3.19×10^{-18}	5.16×10^{-17}	7.42×10^{-17}	1.20×10^{-15}
1.0V	7.61×10^{-18}	1.23×10^{-16}	9.19×10^{-17}	1.49×10^{-15}

[a]

[a] and the correspondingly scaled dimensions for the p-channel MOSFET devices

for the case of cryptographic circuits (as is within the scope of this thesis), the SNR matters. Since the I_{DD} reduces a lot in the sub-threshold region, leading to an overall decrease of both the useful signal and the noise causing an overall decrease of the SNR. Furthermore, we may also conclude that external countermeasures could be added to compensate for the lower values of observed IPN.

- Doubling of the V_{DD}, leads to a quadrature increase of the noise (at lower voltages specifically). This leads to an interesting observation, namely, having multiple levels, (such as in randomized voltage supplies or body-biasing level), could indeed, be suitable for increasing the output noise current.

- The impact of the increase in the number of active elements (i.e., MOSFETs) on the integrated noise σ^2 value is clearly observed.

- From Table 4.5, we compute the integrated noise sigma and obtain similar values (of the same order of magnitude; for instance, the $\sigma^2_{\mathrm{I_{DD}},XOR}$ for $V_{\mathrm{DD}} = 0.9$ V) using a `.ac noise` based analysis. This allows us to back-correlate the values obtained in the previous paragraphs (which were similarly obtained via separate computations of the dynamic and static regions using a *transient noise* based analysis).

Further, we analyze the impact of the K_f factor on the intrinsic noise, σ^2 [A^2] (computed by the integral over the $\mathrm{S}_{I_{DD}}$over the frequency

range). From Table 4.6 and Table 4.5, we observe similar trends for a 2-bit XOR gate. In case of the XOR, for a similar increase of the K_f factor, for e.g., 10^3, an increase of two decades of the noise is observed, similar to observed in Table 4.1 during the analysis of the K_f factor on a single MOSFET; keeping in mind that the array for transistors (in the XOR gate) are now no longer limited to a single RVT only; rather they form a complex transfer function whose output (noise) is dominated by the bandwidth of such a circuit. For the same reason, thanks to the "filtering" setup encountered in such designs (provided by the combination of the linear resistances and an output capacitance), we expect a gradual filtering of the noise output. However, due to the increase in the number of devices and correlations between them, the overall effect is dominated by the increase in noise.

Table 4.6: Impact of K_f parameter on the Noise for a 2-bit XOR at V_{DD} = 0.9V; $V_{\mathrm{G}} = V_{\mathrm{DD}}/2$; $V_{\mathrm{D}} = V_{\mathrm{DD}}/2$ under `.ac noise` simulations

Flag	Corresponding K_f = 10^nsig_dnoise	$S_{I_{DD}}$, 1 Hz [A^2/Hz]	σ^2, [A^2]
nsig_dnoise = 0	1	7.42×10^{-17}	1.23×10^{-15}
nsig_dnoise = 1	10	3.51×10^{-16}	5.68×10^{-15}
nsig_dnoise = 4	10000	3.77×10^{-14}	6.09×10^{-13}
nsig_dnoise = 10	10^10	4.36×10^{-10}	7.05×10^{-9}
nsig_dnoise = -4	0.0001	1.57×10^{-19}	4.23×10^{-18}

4.5 Conclusions

In this chapter, we introduced the reader to the TIMESPAN methodology which allows us to evaluate and quantitatively estimate the *intrinsic device physical noise* existing in any cryptographic implementation. We have made the reader familiar with the necessary tools and techniques which will be essential for appreciation of the validation of the aforementioned methodology in the subsequent two chapters. Additionally, we also provide a first-hand introduction of the choices of the simulational parameters which are necessary for carrying out the simulations and their corresponding rationale. While convergence of the computed values is essential for drawing meaningful comparisons, the simulational costs and trade-offs need to be considered carefully to avoid extremely long simulation times and over-runtime CPU usage, which lead to rather a negligible (or at most a marginal increase) in the estimated values. Further, we have familiarized the reader with a validation of the methodology introduced from chapter 4 through the workings of a single transistor (nFET) device and a 2-bit XOR (as an example of a basic digital building block in

cryptography). We further, point out to the reader statistical properties of the current flowing through an XOR and the challenges encountered in accurately modeling the dynamic region of such traces. We further present techniques to increase the total noise of such basic primitives and their subsequent impact on the supply voltage variations.

Chapter 5

Transient Noise Analysis - II

In this chapter, we implement our introduced methodology on the workings of larger cryptosystems. We show the scalability of our methodology on the workings of more complex digital logic blocks (presented here using the example of 3-bit Sboxes); moving to more complex designs such as the PRESENT and AES Sboxes and finally comparing the AES Sbox-based CMOS and Dual-rail logic (noisy) implementations.

5.1 Case Study 3 : Digital Logic Blocks

In this section, we detail a further implementation and subsequent validation of our proposed methodology building on larger building blocks. The goal is to analyze the behavior of the IPN as it scales up with the number of transistors. Naturally, this leads to a more complex and non-linear system as compared to a single RVT MOSFET or a 2-bit XOR.

5.1.1 Target Designs

Before moving onto designs comprising larger S-boxes (for instance, the 8-bit AES Sbox), we analyze the intrinsic noise characteristics and the application of our introduced methodology on a 3-bit Sbox (comprising of 84 transistors) (Figure 5.1) [4] and a parallel implementation of the 3-bit Sbox consisting of 3 stages of the 3-bit Sbox from above (comprising of 330 transistors). Furthermore, the results from these simulations could provide some insights for the design of masking schemes which (may) utilize such parallel implementations All the circuits are designed in 28nm FDSOI technology using Cadence Virtuoso tool with minimum sized MOSFETs to maximize the noise sources (specifically, the flicker

noise, ref. Equation 2.14) and have the corresponding number of buffered inputs and fan-outs respectively. The Sboxes are provided with a separate V_{DD} to avoid any mixing of the signal coming from the input/output buffers. The current at the V_{DD} supply is measured for calculating the "cryptographic" signal and noise.

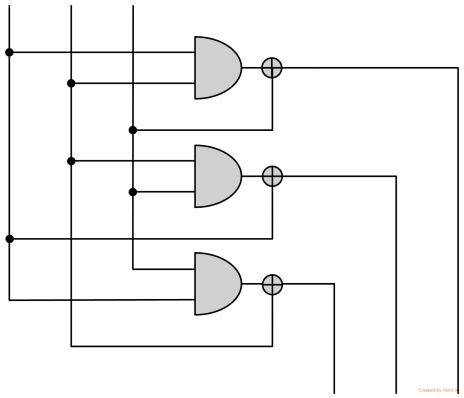


Fig. 5.1: A 3-bit Sbox

5.1.2 Simulation settings

All the 3 designs are simulated with the Transient Noise analysis built in `Eldo` (called by the `.noisetran` command) upto 100 transient noise runs. The noise sources correspond to the physical flicker and thermal noises intrinsic to the MOS transistors. They are generated by `Eldo` in the frequency bandwidth specified by the input parameters of the transient noise analysis. In our first case, we chose $f_{min} = 0Hz$ and $f_{max} = 1GHz$ to include noise sources over the full circuit bandwidth. The input data signals to the circuits are a recurring 0 to an arbitrary input for 4 transitions (for the 2-input XOR) and 56 transitions (for the 3-bit Sbox and parallel implementation). All the input transitions, both for the 3-bit Sbox and the Parallel 3-bit Sbox occur simultaneously; moreover, they all transition to the same value to maintain coherency among the different stages[1]. All simulations are done at 298°K and for a V_{DD} range from 0.3V to 1V. Next we will analyze the effect of the following .noisetran parameters (i.e. maximum frequency of noise source, CPU runtime and number of transient runs) on the "crypto" signal and noise of the XOR gate. Two important simulator choices need to be

[1] the goal here is not to correlate the data-dependency of the transitioning inputs, for which different combinations could have been a better choice; rather the focus here is to show the evolution of the noise as the total number of transistors is scaled up

made, the number of transient noisy runs (`nruns` parameter) and the maximum frequency of the noise generating sources (`FMAX` parameter). The choice of these two parameters is certainly not arbitrary and forms the basis of our methodology where we provide a framework of how to choose these two important parameters which affect the Figures-of-Merit (FoMs) and the CPU runtime.

5.1.3 Results of the Transient Noise Analysis

In this section, we now discuss the results of our ongoing work. We aim to show our present results for a 2-input XOR, a 3-bit Sbox and a parallel implementation of the 3-bit Sbox.

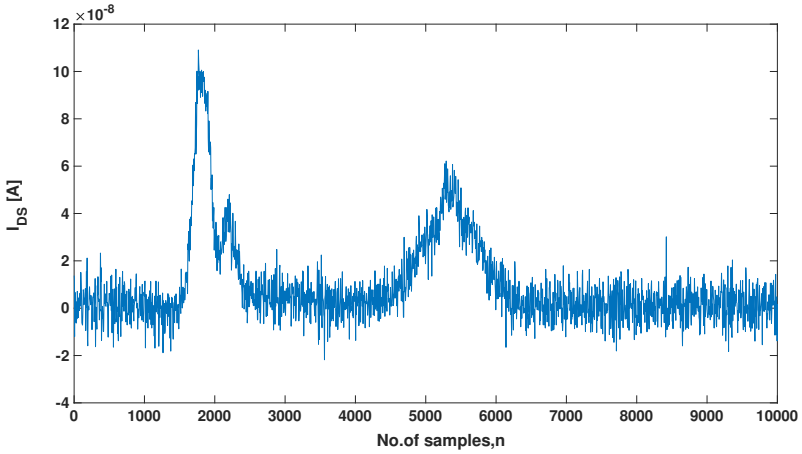


Fig. 5.2: Current trace of a single input transition for a Parallel 3-bit Sbox V_{DD}=0.3V

Similar observations can be stated on moving from a 2-bit XOR (12 transistors) to a parallel implementation of the 3-bit Sbox (330 transistors). Figures 5.2 and 5.3 show the linear current trace and crypto parameters for the parallel implementation for V_{DD}=0.3V. Similarly, Figures 5.4 and 5.5 show the linear current trace and crypto parameters for the parallel implementation for V_{DD}=0.9V. In both the cases, i.e for the 2-bit XOR and 3-bit Sbox parallel implementation, the trend remains the same.

5.1.4 Effect on the Cryptographic Signal, Noise and SNR

From the previous discussion, we are now able to choose the `.noisetran` parameters and fix them for our next analysis. In this section we discuss

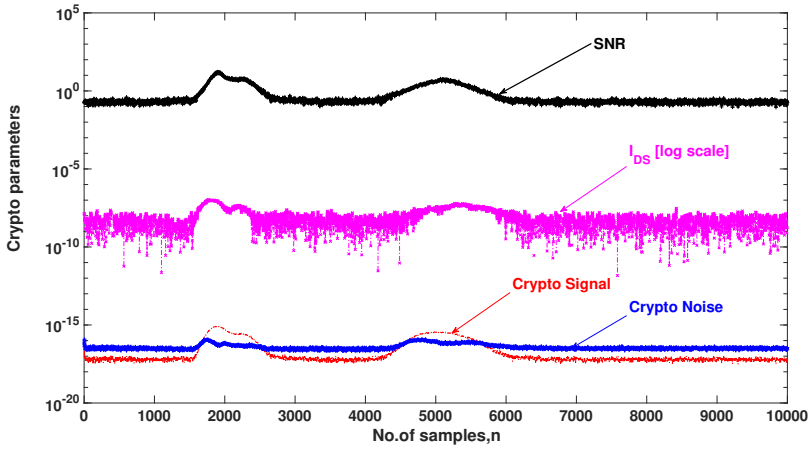


Fig. 5.3: Crypto parameters for a Parallel 3-bit Sbox V_{DD}=0.3V

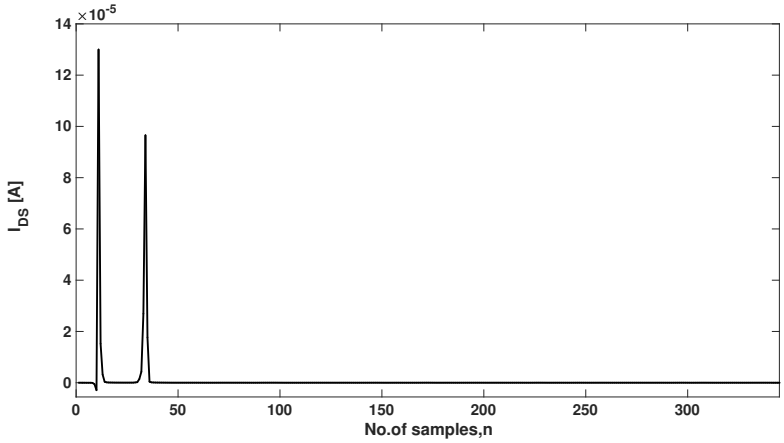


Fig. 5.4: Current trace of a single input transition for a Parallel 3-bit Sbox V_{DD}=0.9V

the impact of the cryptographic signal, noise and the SNR w.r.t the supply voltage, V_{DD} and the number of transistors, N_T. Thanks to the 40 transient noise runs and choosing $fmax$= 1GHz, we are now able to calculate the maximum signal and the maximum noise, (as defined by Equation 2.30). Figure 5.6 shows the scaling of the maximum signal and the maximum noise for a range of V_{DD}, i.e, from 0.3V to 1V.

This allows us to make the following observations:

1 By reducing the supply voltage, the signal and noise values decrease, which is an expected trend. This can be explained by the fact

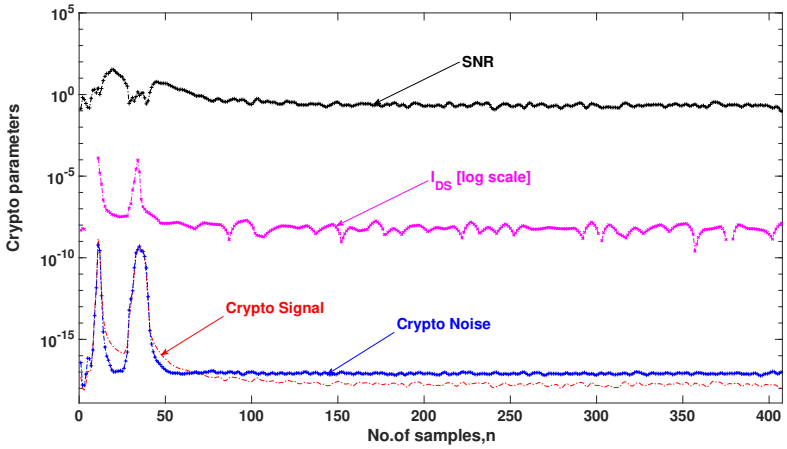


Fig. 5.5: Crypto parameters for a Parallel 3-bit Sbox, V_{DD}=0.9V

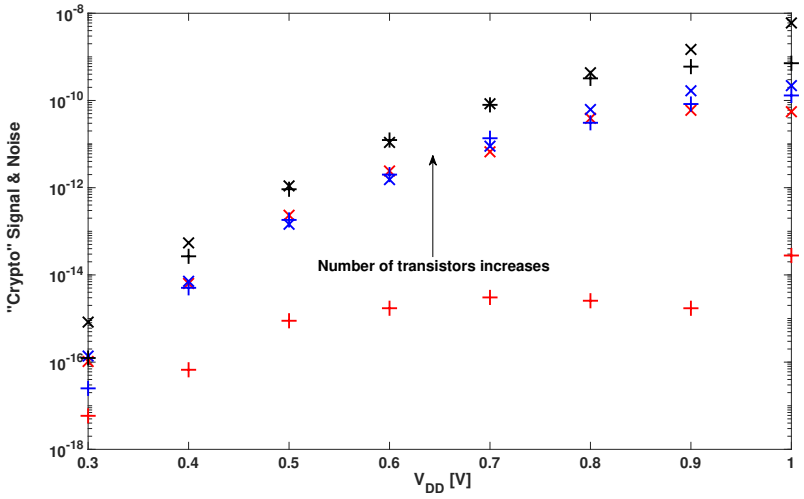


Fig. 5.6: Scaling of "Cryptographic" $\times Signal$ and $+Noise$ as a function of V_{DD} for 2-bit XOR, 3-bit Sbox and Parallel 3-bit Sbox

that as V_{DD} increases, the value of the corresponding I_{DD} also increases which increases the value of the signal. Also, since the noise increases with the increase in V_{DD} (see Equation 4.30), the corresponding noise current also increases.

2 For a particular V_{DD}, the signal (as well as the noise) increases with the increase in the design complexity, i.e with the number of

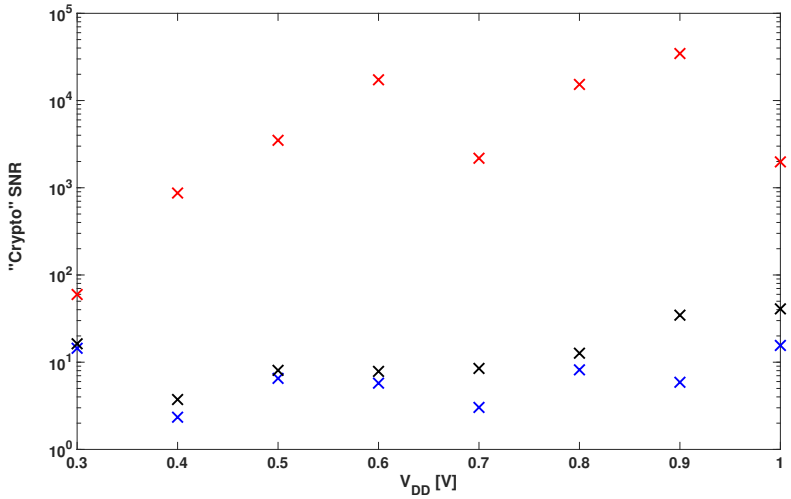


Fig. 5.7: Scaling of "Cryptographic" *SNR* as a function of V_{DD} for 2-bit XOR, 3-bit Sbox and Parallel 3-bit Sbox

transistors. (shown by arrow in figure 5.6). This is also an expected trend as the number of transistors increases, i.e moving from a 2-bit XOR (12 transistors) to a 3-bit Sbox (84 transistors) and finally to a parallel implementation (330 transistors), the signal increases and more noise is accumulated. Hence, the increase in the signal and noise values.

3 The SNR, in Figure 5.7, is in sync with the signal and noise values described above. For the 2-input XOR, the SNR is higher than the 3-bit Sbox (and the parallel implementation) which is expected because of the lesser number of transistors. For the case of 3-bit Sbox and parallel implementation, the SNR's are close to each other suggesting that there is a possible flattening due to the corresponding increase in both the signal and noise values (aided with the increase in the number of transistors) (in the same order of magnitude, of course).

5.1.5 Impact of the *.noisetran* parameters on the simulation cost

Since the `.noisetran` analysis specifies a number of input parameters, it is of importance to analyze the effect of each of these on our calculated values of signal and noise. We first analyze the effect of the f_{max} parameter specified in the `.noisetran` analysis, as shown in Figures 5.8 and 5.9.

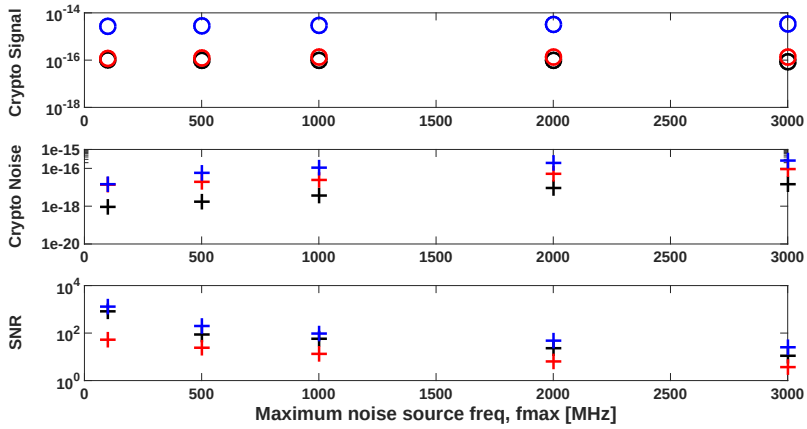


Fig. 5.8: Convergence of f_{max} parameter on the "Cryptographic" *Signal*, "Cryptographic" *Noise* and the *SNR* for a 2-bit XOR, 3-bit Sbox and Parallel 3-bit Sbox for V_{DD}= **0.3**V with nruns=100

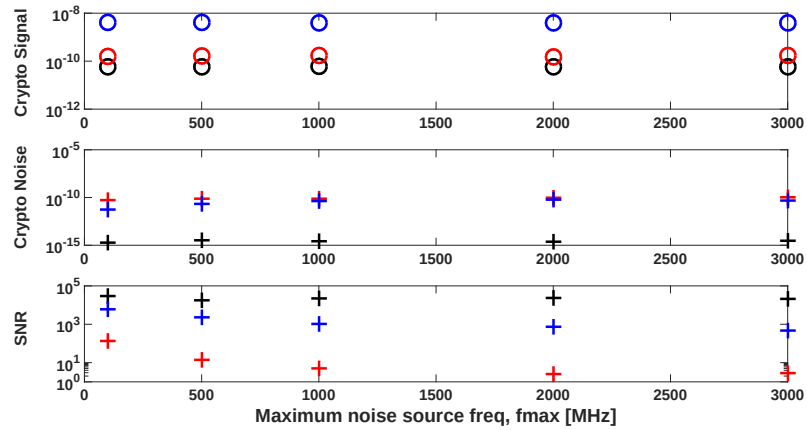


Fig. 5.9: Convergence of f_{max} parameter on the "Cryptographic" *Signal*, "Cryptographic" *Noise* and the *SNR* for a 2-bit XOR, 3-bit Sbox and Parallel 3-bit Sbox for V_{DD}= **0.9**V with nruns=100

Choosing a larger f_{max} also means increasing very significantly the CPU runtime, as shown in figure 4.4(a). We then propose to choose an optimum value of the f_{max}, which trades-off CPU simulation time and constant signal/noise values. As seen in Figures 5.8 and 5.9, for our simulations, choosing a value of upto 1000 MHz ensures that the values of signal and noise calculated remain essentially unchanged (and do not fluctuate by an order of magnitude). We recall the observations made in

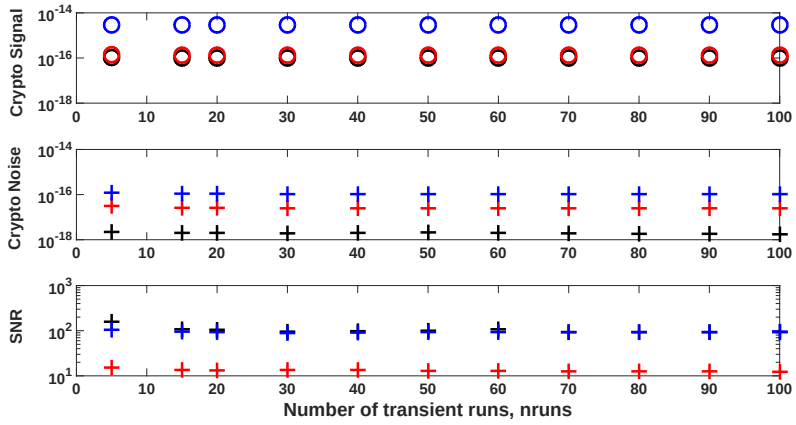


Fig. 5.10: Convergence of *nruns* parameter on the "Cryptographic" *Signal*, "Cryptographic" *Noise* and the *SNR* for a 2-bit XOR, 3-bit Sbox and Parallel 3-bit Sbox for V_{DD}= **0.3**V with fmax=1000 MHz

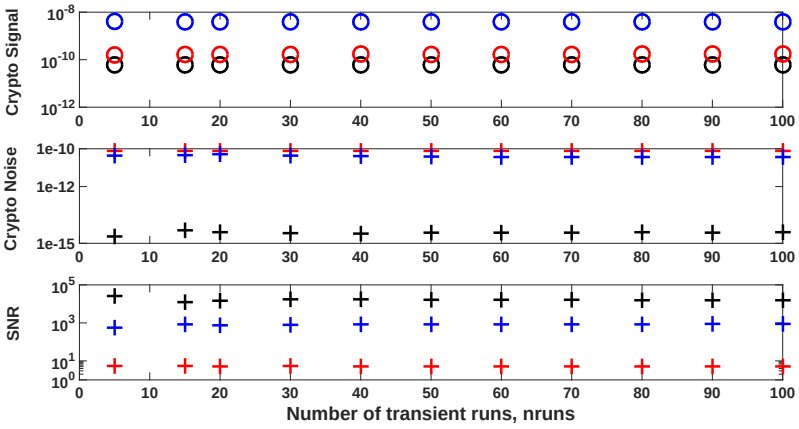


Fig. 5.11: Convergence of *nruns* parameter on the "Cryptographic" *Signal*, "Cryptographic" *Noise* and the *SNR* for a 2-bit XOR, 3-bit Sbox and Parallel 3-bit Sbox for V_{DD}= **0.9**V with fmax=1000 MHz

Section 4.2.2, using Equations 4.11 and 4.12 and (re)-invoking Figures 4.3 and 4.4, where we see a quasi-linear dependence of the CPU runtime with the increase in `FMAX` at no additional gain of the security metric, i.e., the SNR in this case. Choosing a larger f_{max} value hence would only lead to longer simulation times and would only increase with the design complexity providing no additional benefit. Hence, an optimum value of f_{max} must be chosen which minimizes the simulation time. The

observations from Figures 5.8 and 5.9 here further validate our claims concluded from the previous chapter and further hold true, even as design complexity is increased.

Next we analyze the impact of the number of transient runs, `nruns`, on the simulation time. The greater this value, longer the simulation time; for our simulations, we chose a range of `nruns` values and calculate the "cryptographic" signal and noise and observe (Figures 5.10 and 5.11) that it essentially remains constant as the number of transient runs increases from 5,10,20,.....to 100 runs in intervals of 10. In this case, the optimal value of choosing `nruns` = 40 runs with a correctly chosen f_{max} value to minimize the CPU runtime (cost) keeps provided a convergence of the computed values within the tolerance limit (i.e., respecting Equations 4.11 and 4.12).

5.2 Case Study 4: Cryptosystems

In this section, we discuss the results of our `transient-noise simulation` methodology over different implemented design i.e., for a 2-input XOR, a 4-bit Sbox of the PRESENT encryption and an 8-bit AES encryption Sbox. These include a total of 12, 684 and 1884 transistors, respectively. The goal of this case study is to show the evolution of the noise with increasing circuit complexity.

5.2.1 Target Designs

Our results are based on transient noise simulations in an Eldo environment while using a 28nm FDSOI PDK (process design-kit) provided by an industrial foundry. This PDK is characterized to provide advanced UTSOI v2 [95] noise-simulation abilities of all the noise-parameters discussed above (and many more). In most experiments performed in this research, the sizing of the transistors is kept minimum to maximize the noise produced (flicker noise especially which is reversely proportional to gate-area). We use a simple 2-bit XOR, a 4-bit PRESENT Sbox and an 8-bit AES S-box to show scalability of our results in a relevant cryptographic context. All of which were custom designed using Cadence Virtuoso software, to show the scaling trends of the signal and noise w.r.t the supply voltage and the number of transistors.

5.2.2 Simulation Settings

The input data signals to the circuits covers of All the 3 designs are simulated with the Transient Noise analysis built in `Eldo` (called by the

`.noisetran` command) upto 40 transient noise runs. The noise sources correspond to the physical flicker and thermal noises intrinsic to the MOS transistors. They are generated by `Eldo` in the frequency bandwidth specified by the input parameters of the transient noise analysis. In our case, we chose $f_{min} = 0Hz$ and $f_{max} = 1GHz$ to include noise sources over the full circuit bandwidth. The input data signals to the circuits are a recurring 0 to all possible arbitrary digital input transitions (i.e. 4, 16 and 256 transitions for the 2-input XOR, 4-bit PRESENT and 8-bit AES S-boxes, respectively) at a clock frequency of 10 MHz. All simulations are done at 298°K, TT (typical-typical) process corner and for a supply voltage V_{DD} range of 0.5V to 1V.

From the previous section, two important simulator choices have been made, namely, the number of transient noisy runs (`nruns` parameter) and the maximum frequency of the noise generating sources (`FMAX` parameter). The choice of these two parameters, as has been shown in much detail, is certainly not arbitrary and forms the basis of our introduced methodology where we provided a framework on how to choose these two important parameters which subsequently affects the Figures-of-Merit (FoMs) and the CPU runtime. For subsequent simulations, we fix these parameters as `nruns`= 40 and `FMAX`= 1 GHz, unless otherwise explicitly mentioned.

5.2.3 Evaluating Crypto FoMs-Signal, Noise and SNR from Transient Noise Analysis

After collecting traces from our repeated transient noise runs, we are able to calculate the maximum signal and the maximum noise, (from Equation (2.30)). Figures 5.12 and 5.13 show the scaling of the maximum signal and noise for a range of V_{DD}, i.e, from 0.5V to 1V. We can make the following observations from the 2 plots,

1 By increasing the supply voltage, V_{DD}, the value of the maximum signal and noise increase. This can be explained by the fact that as the V_{DD} increases, the dynamic power consumption, P_{dyn} increases (hence the increase in on current, I_{ON}) which increases the *Signal* value. This is true for the static power leakage as well. The increase in the "crypto" *noise* can be explained by the increase in the thermal noise which increases with the increase of I_{ON}. Reciprocally, signal and noise decrease with V_{DD}. We can observe that the signal decrease is faster than the noise decrease which is of interest for our purpose. A possible reason could be due to the square dependence of the Signal on the supply voltage, $\mathrm{V_{DD}}$ whereas the noise behaves in a more linear way with respect to the $\mathrm{V_{DD}}$.

2 For a particular V_{DD}, the signal (and the noise) increase with the increase in design complexity, i.e as the number of transistors increases for a given circuit (e.g. moving from a 2-bit XOR to a 4-bit PRESENT Sbox to an 8-bit AES Sbox).

The increase in the signal can be modeled by the following relation [87]

$$\bar{S}_{V_{DD}}^{circuit} = S_{V_{DD}}^{XOR} N_T^{\beta} \tag{5.1}$$

where $\bar{S}_{V_{DD}}^{circuit}$ is the signal for the target circuit, $S_{V_{DD}}^{XOR}$ is the signal produced by a 2-bit XOR for the same supply voltage, V_{DD}, N_T is the ratio of the increase in the number of (switching) transistors w.r.t a 2-bit XOR and β is a technology factor which varies $0.4 < \beta < 1.5$ for our circuits and depends on the value of the supply voltage, V_{DD}.

The increase in the noise can be modeled as [87]

$$\bar{N}_{V_{DD}}^{circuit} = N_{V_{DD}}^{XOR} N_T^{\alpha} \tag{5.2}$$

where $\bar{N}_{V_{DD}}^{circuit}$ is the noise for the target circuit, $N_{V_{DD}}^{XOR}$ is the noise calculated for a 2-bit XOR at the same V_{DD}, N_T is the ratio of the number of transistors w.r.t a 2-bit XOR and α is a parameter which scales with the supply voltage, is $\lessapprox 2$ for our circuits and depends on the value of the supply voltage, V_{DD}. Consequently, we observe that noise increases faster with the number of transistors than the signal. This could be related to the fact that the intrinsic MOSFET noise sources are not correlated and hence the total contribution to I_{DD} is larger, whereas the signal is more proportional to the number of (switching) circuit branches connected to V_{DD}.

Since the calculated "cryptographic" noise is essentially a mean of the variance across different inputs for *nruns* number of traces, we should be able to relate this noise to the histogram of the measured current. We see this in the next section. From Figures 5.12 and 5.13, we see the evolution of the maximum signal and maximum noise (which conclude in the computed SNR (2.30)); however, in typical *side-channel attacks*, the adversary, chooses a *Point-of-Interest* (POI) in time and exploits this point to extract the relevant information. For our following results, we choose the maximum SNR time point as our POI. This is intuitive, as the maximum information leakage occurs at this point. Hence analyzing the crypto metrics at this point and plotting the relevant figures-of-merit, such as the SNR or MI, provides us with useful information about the extent of information leakage. [2].

[2]We also use this POI for Mutual Information analysis, as explained

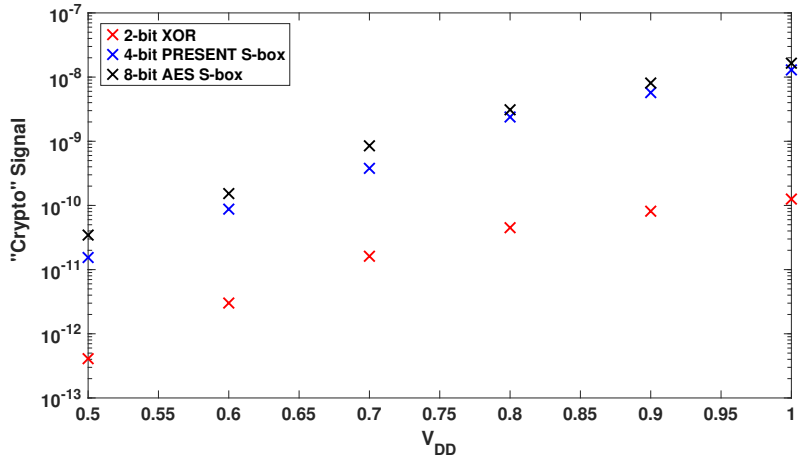


Fig. 5.12: Scaling of "Cryptographic" *Signal* as a function of V_{DD} for different circuits

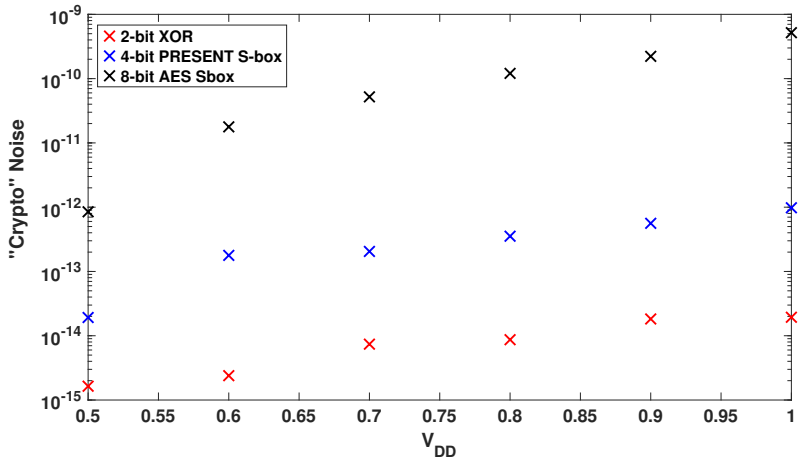


Fig. 5.13: Scaling of "Cryptographic" *Noise* as a function of the number of transistors, N_T for different V_{DD}

In Figure 5.14, we plot the "cryptographic" calculated *Signal*, "cryptographic" *Noise* and the "crypto" *Signal-to-Noise* (SNR) ratio at the *Point-of-Interest (POI)*[3] for different supply voltages, *i.e.,* from 0.5V to 0.9V. The results obtained are consistent with the expected trends;

[3]we note here the difference between figures 5.12, 5.13 and figure 5.14. the former ones plot the *Maximum* Signal and Noise whereas the latter plots the Signal and Noise at the POI

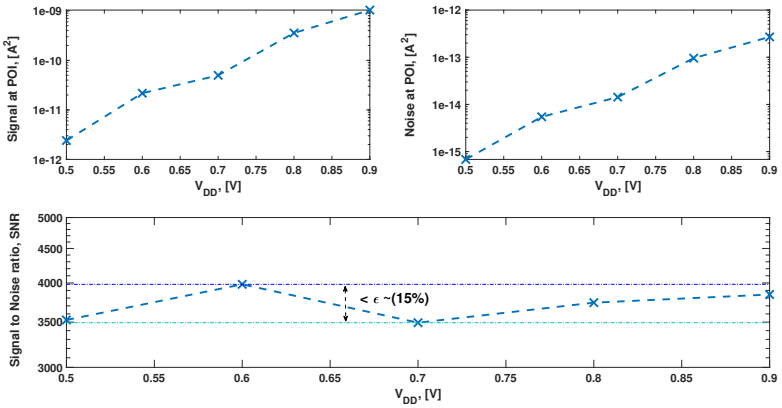


Fig. 5.14: Scaling of "Cryptographic" *Signal*, *Noise* and *SNR* at *Point-of-Interest (POI)* as a function of V_{DD} for AES Sbox

i.e., the signal and noise values increase with the increase in the value of the supply voltage V_{DD}. Concretely, the SNR values also increase with the increase in the supply voltage. However, we categorically state that the SNR values with the V_{DD} are still well within the tolerance limits assumed by our methodology (for instance, the difference between the values of the SNR at 0.5V (value of 3529) and 0.9V (value of 3845)) are well within the ϵ tolerance limit specified[4]; subsequently, this does not allow us state that the assumed trend has indeed been violated. Rather, we state that since the SNR is in itself a first-order metric and is often used by designers only in very early design stages to obtain a very first-hand estimation of the security level; for further and clearer understanding of the impact of the noise variance and its impact w.r.t the supply voltage, we refer to the analysis using the Mutual Information, introduced in section 2.5.5.

Nevertheless, we account for this due to the limited number of traces which were simulated compared to the larger number of traces available *vis-a-vis* from measurements, which often leads to a much smoother trend.

5.2.4 AC Analysis of an 8-bit AES S-box

Figure 5.15 plots the power spectral density of the current (noise current power spectral density) for a given (constant) bias for an 8-bit AES

[4]we remind the reader a tolerance limit, ϵ, of 15% for our methodology as previously introduced in the preceding chapter and used throughout this thesis

Sbox[5]. The inputs are set to a constant bias ($V_{DD}/2$ in this case)[6] and the frequency is swept from 1Hz to the frequency-of-interest. The transistors in the 8-bit AES Sbox have a constant fixed input bias corresponding to $V_D = V_G = V_{DD}/2$; the rationale behind this choice of bias inputs has already been well explained in section 4.4.3.

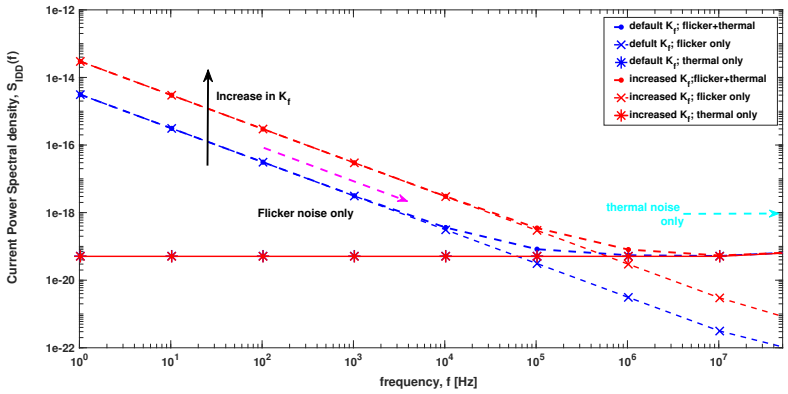


Fig. 5.15: AC analysis of increased K_f at V_{DD}=0.5V, $V_D = V_G = V_{DD}/2$; for 8-bit AES Sbox

From Figure 5.15, we plot the $S_{i_{DD}}$'s for given K_fs. The lines represent the $S_{i_{DD}}$, for the intrinsic K_f, i.e the default K_f value. We then compute the integrated noise variance, $\sigma^2_{I_{DD}}$ thanks to Equation 4.15 from the given $S_{i_{DD}}$. We also observe that by increasing the K_f, we obtain the noise curve (shown by the lines) which clearly shows an increased level of noise due to an increased K_f.

5.2.5 Evaluation using Mutual Information -An Analysis

In this subsection, we revisit our previous results and explore them using the Mutual Information metric introduced in section 2.5.5 for the largest design under investigation, the AES Sbox. As previously explained, the MI metric allows the designer to *quantitatively* estimate the *information leakage* from an implementation and estimate the noise variance needed to achieve a given security level. Concretely, it let us derive a more general intuition and the effect of parameters such as the noise variance and the supply voltage on the information leakage and how the above

[5]to provide a better understanding of a more practical scenario, only the Sbox is considered here to correlate with previously obtained transient noise values

[6]hence the considered current is the static leakage, whose statistics have already been described

parameters vary the MI. Note that the SNR curve is shown for a given number of *nruns*, that is, the SNR as is shown in Figure 5.14, does not take into account the noise averaging. In turn, increasing *nruns*[7] reduces the noise and increases the attack success rate. The MI metric, however, does take *nruns* into due consideration. The MI curves enable an asymptotically security evaluation, i.e. it provides a security trend as a function of the noise or the noise-removal/averaging abilities of the adversary. It also provides the designer an upper bound of the noise adding capability to reduce the leakage to achieve a desired security level. The * markers in Figure 5.17, represent the inherent device physical noise that an adversary cannot further remove (as compared to external/ environmental/ algorithmic noise [99]).

Increasing the K_f factor-Revisiting using MI

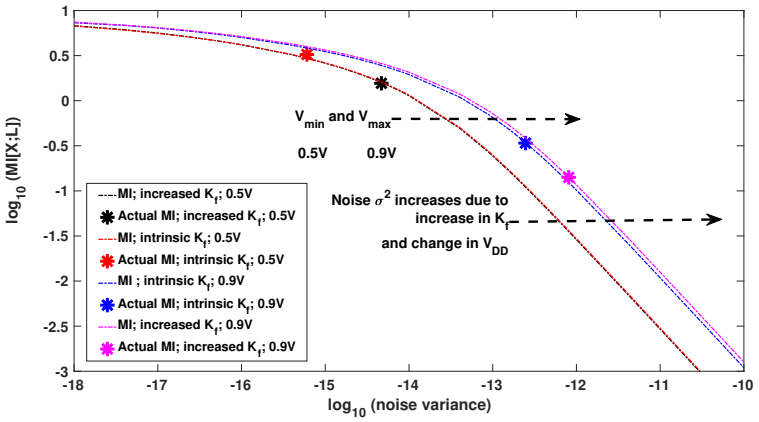


Fig. 5.16: Effect of the increase in the K_f factor for a CMOS AES Sbox at $V_{DD} = 0.5V$ and 0.9V. The individual markers $*$ correspond to the actual calculated noise variance from the simulator.

In this subsection, we revisit our discussion on the K_f factor in the *context of Mutual Information*. Using *transient noise simulations*, we observe from Figure 5.16, the results of the increase in flicker noise *on the MI* for a CMOS AES Sbox (same setup as above), for a supply voltage $V_{DD} = 0.5V$ and 0.9V. To understand the curve, we must first be clear about the legend of the figure. The dashed line indicates the MI due to the *intrinsic* K_f, i.e., K_f which has not been modified and is

[7] i.e, averaging over a *very* large number of noisy traces, typically a *million*, in practical side-channel attack scenarios

the default as provided by the foundry model. The black dashed curve indicates the MI due to the *increased* K_f, i.e K_f which has been modified by changing the parameters of the foundry model. We make the following observations,

- A decrease in the magnitude of the MI is observed (illustrated using the colored $*$ markers), thanks to the increased K_f. As the K_f increases, so does the noise variance and the MI goes down. This is observed for both the supply voltages plotted, i.e 0.5V and 0.9V
- We also observe that the noise variance value is higher for the higher supply voltage, i.e 0.9V compared to 0.5V. This is an expected trend and validates our results obtained from the Signal and Noise curves above.
- Furthermore, the MI curves are able to clear the discrepancy observed in Figure 5.14 regarding the (un)-clear trend of the SNR with respect to the V_{DD}. At this point, we can clearly conclude that although increasing the supply voltage increases the value of the signal and noise metrics, overall the effect is dominated by the increase in the noise, and hence the information leakage (computed by the SNR or the MI) is expected to decrease.

From Figure 5.15, we obtain actual noise values computed from the noise spectral density (integrating over the frequency limits `fmin` to `fmax`). The noise computed in case of cryptographic analyses, from the cryptographic FoMs (for e.g., the SNR or MI), is indeed calculated from the transient noise simulations (using the methodology introduced). By extending the noise computed from a single transistor to an AES Sbox, we provide an intuition into the security levels of a given implementation taking IPN into account. Indeed, the same degree of increase (by an order of magnitude) in K_f is obtained using both the MI curves, i.e., ref. Figure 5.16 and Figure 5.15.

Effect of Supply Voltage on the Mutual Information

In this section, we revisit the results from Figure 5.14 in the context of Mutual Information and provide a more detailed analysis of the impact of the supply voltage on the amount of information leakage. From Figure 5.17 we make the following observations,

- Reducing the voltage leads to a signal reduction which in turn reduces the MI (for a given noise level). This is clearly illustrated

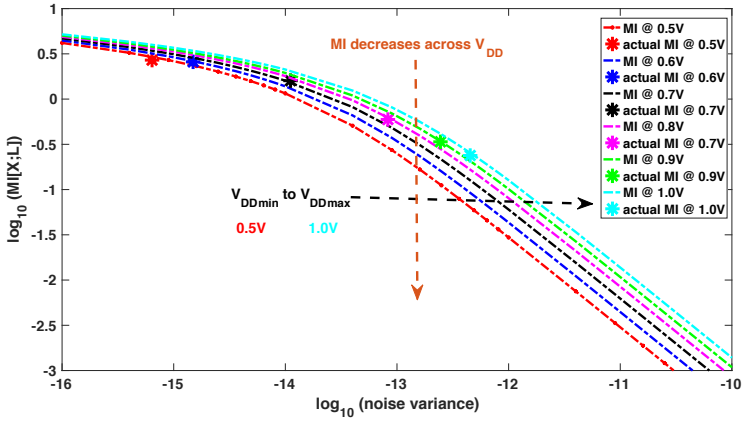


Fig. 5.17: Impact of the Supply Voltage, V_{DD} on the MI for a CMOS AES Sbox. Devices here have been simulated at the default K_f provided by the model. Individual markers $*$ correspond to the actual calculated noise variance from the simulator.

and is an expected trend. This is an interesting result as it makes clear that lowering the supply voltage does not always necessarily lead to (*more*) leakier implementations, if intrinsic physical noise is taken into account.

- The noise variance increases with the supply voltage. This is consistent with the results obtained using the SNR metric in the previous section which are expected. This allows the designer to estimate the noise required to be added for a given supply voltage to achieve a desired security level.

- Using the actual computed MI values and noise (represented by $*$ in Figure 5.17) the corresponding noise variances obtained are in close agreement with the noise variances obtained from transient noise simulations (SNR) and AC analyses, as described in the previous section.

A note on the MI vs SNR Although theoretically, from Equation 2.32, lower SNR values correlate with lower values of MI. However, this relation holds (asymptotically) true in case of a very large number of traces (which is almost always the case in real-world measurements). Compared to in a simulations-based scenarios (as is the case here in the context of this thesis), the CPU runtime and memory (storage) complexity of the output file is severely restrictive. Nevertheless, we also

conclude similar trends with the change in the supply voltage (for both the MI and the SNR) with the change in the noise (variance) as clearly illustrated in Figure 5.17.

5.3 Case Study 5: (Noisy) CMOS vs DDSLL

As a final case study to validate the scope of our introduced methodology, we compare the CMOS and DDSLL S-box implementations. This allows us to broadly state that the methodology introduced in this thesis allows for a comparison of larger cryptosystems; more specifically, it allows for computation of security evaluation metrics at design stage. Further more, it opens the scope for further research into optimizations needed to reduce the simulation cost (both in terms of computational complexity and time); this would allows for further granulation of the noise when building larger cryptosystems with multiple blocks.

5.3.1 Target Designs

The target designs for this case-study are CMOS and DDSLL 8-bit AES Sboxes (as described in section 2.5.1). The underling DDSLL gates were implemented (with Cadence Virtuoso) utilizing minimum sized transistors from the 28nm FDSOI PDK. The sizing of the transistors has been kept minimum to maximize the noise (especially the flicker noise). Using the methodology introduced in this chapter, we simulate flicker and thermal noise from the transient noise analysis by Eldo simulator (from Mentor Graphics). The currents drawn from the modules-under-test are recorded from the simulator (over multiple supply voltages), i.e $(I(V_{DD}(t)))$, and the security metrics computed as described in the previous subsection.

5.3.2 Simulation settings

The designs are simulated using the `Eldo` simulator with intra-cell layout characteristics only. The Transient-Noise analysis built in `Eldo` (referenced by the `.noisetran` command) has been repeated upto N_{runs}[8] transient noise simulations. The noise sources correspond to the physical flicker and thermal noises intrinsic to the MOS transistors were generated by Eldo within the frequency bandwidth specified by the input parameters of the transient noise analysis (Equation 2.28).

The chosen minimal flicker noise value, $f_{min} = 1/T$, where T represents the total simulation time for all possible inputs. The input data signals to the circuits are a recurring 0 to an arbitrary input of 256 transitions at a

[8] N_{runs} was chosen as such that the computed variance has converged

clock frequency of 10 MHz and $f_{max} = 1/(2*dt)$, where f_{max} represents the maximum frequency of the noise generating sources, dt is the minimum time step being used by the simulator or specified by the user and as such that increasing it did not increase the signal variance. This was done to reduce expensive simulation time (as explored in [87]). All simulations are done at 298°K, TT corner and for a V_{DD} range from 0.5V to 0.9V

5.3.3 Transient Noise Analysis: Results

In this section, we analyze the results of our transient noise analysis using the security metrics described in section 2.5.5. The examined first order univariate metrics provide a first hand early stage security level of a cryptographic implementation. In this work, we do not specify a target SNR or MI value; rather, our goal is to exploit the methodology introduced for dynamic dual-rail logic level implementations and show that the amount of noise needed to be added to reach a given security level is smaller when fully (and correctly) analyzing the intrinsic noise.

CMOS vs DDSLL- Analysis using the SNR

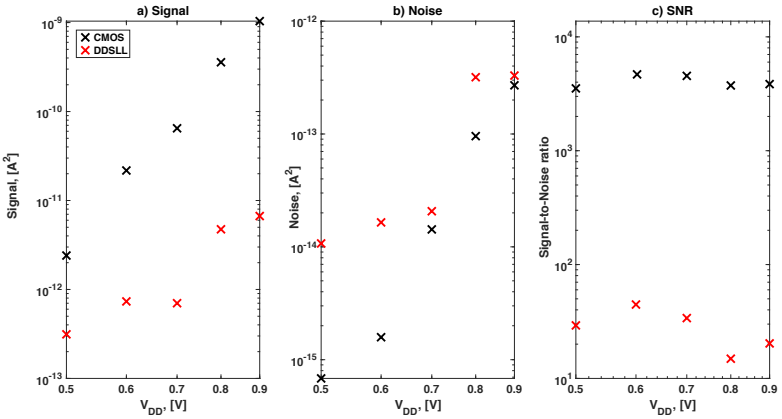


Fig. 5.18: Impact of the V_{DD} on "Cryptographic" *Signal*, *Noise* and *SNR* for CMOS & DDSLL Sboxes across V_{DD}=0.5V to V_{DD}=0.9V

1 We first analyze the impact of the *Signal* metric as shown in Figure 5.18a. We make the following observations from this,

the Signal increases with the increase in the supply voltage, V_{DD}, for both CMOS and DDSLL circuits. This is expected as the device *on* current increases with the voltage. In addition, CMOS

leaks higher signal as compared to DDSLL. Further, the difference in slopes of the CMOS and DDSLL are observable and could be attributed to the underlying implementation style, namely, since DDSLL logic operates by using the pre-charge (during which, $V_{swing} = V_{fixed}$)and during evaluation (where the power is now a function of (V_{swing}, V_{DD}), i.e., $V_{swing} = V_{fixed} - V_{DD}$), the overall signal is dominated by the difference in the voltage (ΔV) rather than the actual V_{DD} values. Furthermore, as the devices operate in the weak-inversion (and just above the threshold region) compared to stronger-inversion region operation (i.e., $V_{DD} = 0.7V$ and above), the apparent difference for CMOS is very obvious.

2 Figure 5.18b shows the "cryptographic" noise as a function of the supply voltage for the two designs. We observe the following,

- At a given voltage, V_{DD}, DDSLL has a higher noise impact than CMOS, as more number of transistors are switching in DDSLL[9]. Although Static-CMOS has an overall higher number of transistors (1884 in this case) compared to DDSLL (consisting of 1275 transistors), the number of switching transistors is clearly higher for DDSLL compared to CMOS(which tends to have a lower activity switching factor, α_F (ref. Equation 4.28)[10] [87].
- Second, and more importantly, we see that the noise contribution of DDSLL (as compared to CMOS), is higher at lower supply voltages, e.g. 0.5V-0.6V than at higher voltages say, 0.8V-0.9V. This could be attributed to the operation of the dual rail logic gates in the weak inversion region (i.e, just above the threshold voltage of the device) where the static leakage is still comparable to the dynamic component (i.e., $I_{leak} \cong I_{dyn}$; ref. Equation 2.2).

3 From Figure 5.18(c), we see the trend of the *estimated* SNR; due to its lower signal value and higher noise compared to DDSLL, the DDSLL has much smaller SNR values for all supply voltages.

To conclude, whereas in Chapter 3, only the signal part of DDSLL as a function of the supply voltage was evaluated, here we complement this result by the positive affect of the noise as a function of the supply voltage (and as compared to CMOS). This in turn, translates to an SNR reduction of $\times 2$ orders-of-magnitude across a 400mV voltage span.

[9] thanks to the precharge and evaluation phase

[10] 0.1 - 0.2 in most of the cases

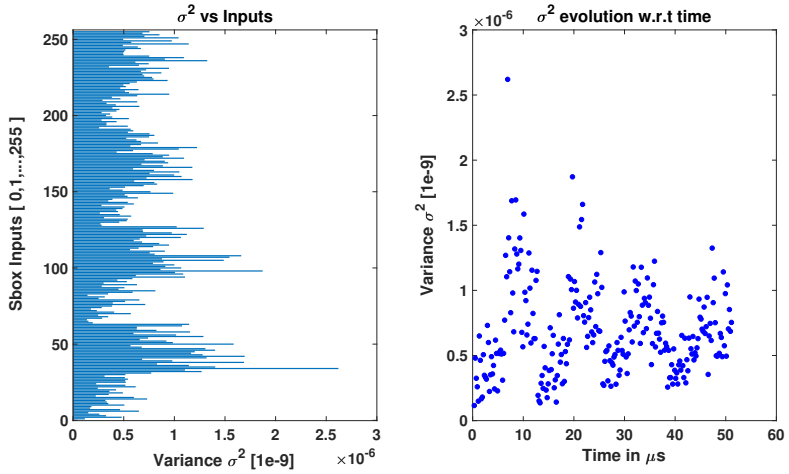


Fig. 5.19: Noise Variance (qualitative) correlation for a CMOS Sbox at V_{DD}=0.5V

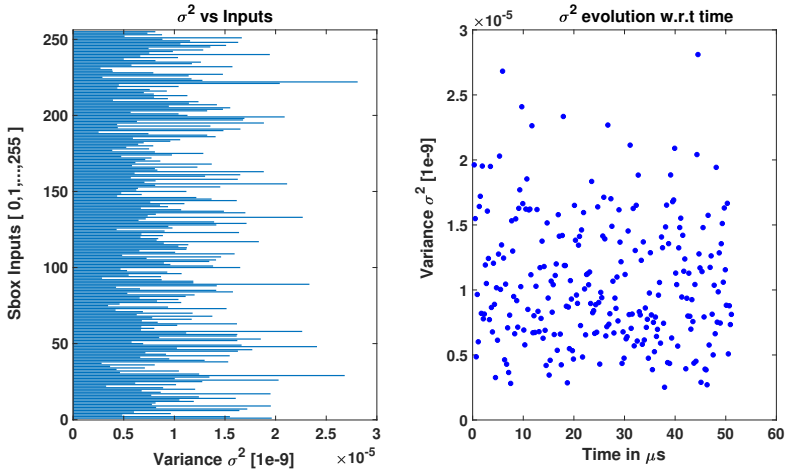


Fig. 5.20: Noise Variance (qualitative) correlation for a DDSLL Sbox atV_{DD}=0.5V

Analysis of noise variance

In this subsection, we visually analyze the *intrinsic*-noise variance. We show that DDSLL presents a stronger noise independence with respect to the inputs as compared to CMOS. Without the loss of generality (and as was observed across supply voltages) for this discussion, we use a supply voltage of $V_{\mathrm{DD}} = 0.5$V. We plot the noise variance computed over $\mathrm{N_{runs}}$ transient noisy runs, for all possible inputs. The variances are plotted as

bar plot as a function of the S-box inputs, and as a scatter plot vs. the simulation time side-by-side (the scatter plot is used to better illustrate the distribution) for CMOS (Figure 5.19) and for DDSLL (Figure 5.20) respectively. We make the following observations,

- Qualitatively, for CMOS, the noise variance is more correlated. The correlation influences the switching factor (i.e., the number of switching MOSFETs and hence influences the signal variance). In addition, the distribution that is observed contains multiple repeated patterns (of correlations). The correlation dependence on the power consumption of a CMOS circuit is well known from literature and already been introduced in the Background section of this dissertation. The power consumption (i.e., the switching current) at a given supply voltage, is a function of the activity factor (i.e., the number of transistors switching ON or OFF, which is correlated with the adjacent temporal samples). Since the noise variance is computed from (switching) current consumption, the this dependence is expected.

- For DDSLL (Figure 5.20), it is observed (qualitatively) that there exists a much uniformly distributed noise variance. e.g. as in contrast to CMOS, the noise (correlation) is more independent. In comparison to CMOS, DDSLL switching transistors are always pre-charged to 'high' logic level; and then depending upon the input, subsequent transistors are "evaluated". Consequently, this provides much lower dependence (i.e., much lower correlation) between the adjacent temporal samples and the transient power (or current) consumption. Nevertheless, we emphasize that physical imperfections still exist and manifest as, for instance, the imbalance between the capactitative loads of the output wires (which can still be seen in Figure 5.20). Although, DDSLL reduces the data-dependencies, but it does not eliminate them completely.

- More importantly, we observe that DDSLL has a higher (qualitative) noise-variance as compared to CMOS by almost an order of magnitude. This reinforces the notion that the noise calculated from Equation 2.30 provides a good estimate for calculating the SNR from transient noise simulations. We explore this in further detail in the next section.

It is important to note that an attractive property of the DDSLL design is the increased independence of the noise from the inputs. In fact, it is a common criteria for masked hardware and software applications.

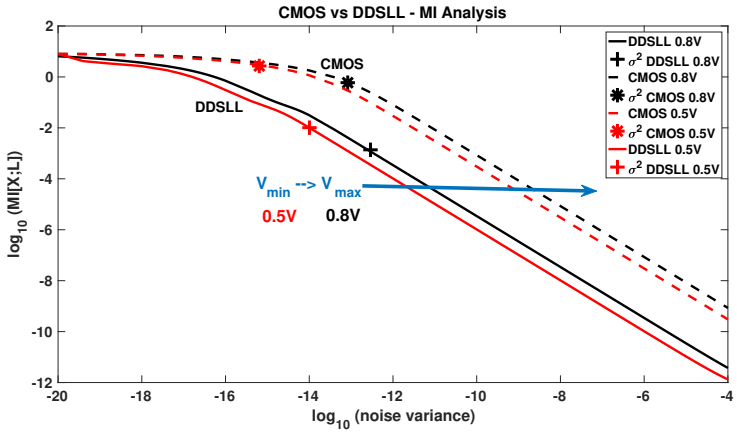


Fig. 5.21: Mutual Information as a function of the intrinsic Noise Variance of CMOS and DDSLL circuits for 0.5V and 0.8V. The individual points correspond to the actual calculated noise variance from the simulator.

Mutual Information Analysis

The MI for CMOS and DDSLL implementations for supply voltages (0.5V & 0.8V) is plotted as a function of the simulated noise variance. The actual noise variances calculated from the transient noisy simulations are shown using markers on the curves. We make the following observations from Figure 5.21:

- For CMOS circuits, the amount of information leakage (MI) is higher than for the corresponding DDSLL leakage for all noise variances across all supply voltages. This confirms our intuition, that although DDSLL loses its advantage w.r.t technology scaling, as discussed in Chapter 2, it (could) continue to be a design-of-choice when physical *intrinsic* noise is taken into account. This advantage is especially attractive at lower voltages.

- Generally, CMOS leaks more information for a given noise level than a DDSLL implementation. Furthermore, a DDSLL implementation inherently has more noise (on account of the larger number of transistors), which contributes to its security.

- It is possible to see that the difference of the MI value of DDSLL between 0.8V to 0.5V is smaller than the same for the CMOS designs, which reassures the conclusions[11] from the previous subsection (i.e.,

[11] that the noise contributions at lower voltages are significant compared to at higher voltages

from ref. Figure 5.18).

CMOS or DDSLL? A nice property of the MI curves is that it enables a designer to estimate the amount of noise required to be added to a design (to make it more secured). In our case, for a reasonable and similar security level (below 10^{-1}), CMOS design will have to incorporate two order-of-magnitude more noise than DDSLL designs.

However, the higher PDP values of DDSLL compared to CMOS tilts the favor back to CMOS based implementations. Recalling, Figure 3.3 in 28nm, where DDSLL performs worse than CMOS, although marginally. The effect is slightly exacerbated at lower voltages of operations and evens out at higher voltages; thus leading a free choice to the designer. Additionally, when considering operations at higher voltages (ref.Figure **??**), DDSLL loses its advantage. To summarize, on account of these factors and considering the design complexity of dual-rail logic designs (the need for balanced capacitances, input and output buffers for voltage level shifting, and the inherent need to design components as symmetrically as possible, in addition to the large and more number of gates needed to even implement a basic functional gate) and its incompatibility with standard digital design flow tools (for e.g., using standard cell libraries), keeps the tilt in favor of CMOS based designs, atleast for the foreseeable future.

5.4 Conclusions and Open questions

In this chapter, through a series of validating case studies, we have proposed a methodology to concretely simulate intrinsic MOSFET physical noise and to link the results to concrete security evaluation metrics of cryptographic modules using transient noise simulations. The results of this research are valuable to discuss and answer questions such as, how the physical noise sources from the MOSFETs can be used for effectively deriving the cryptographic SNR or MI values, and especially to concretely perform security evaluation at design stages of a cryptographic implementation.

Not limited to security evaluations, this work also serve as a design guideline for early stage cryptographic (and digital) designers to incorporate security from the beginning of a product development cycle. Such guidelines include (but not limited to)

1. the design using lower sized transistors (keeping in mind the performance required and respecting the noise margins at the given

voltage of operation)

2. architectural level designs such as to maximize the impact of physical noise (by introducing hierarchy in the levels of implementation, using a mix of say, serial and parallel implementations), and

3. using a higher supply voltage (modes of) operation, if possible, to reduce the impact of the leaking signal (as has been shown already).

This allows designers to trade-off between security and other metrics like area, power consumption, etc. keeping in mind the (intrinsic) vulnerability of such devices to Differential Power Analysis (DPA) [63]. We believe that our introduced methodology not only provides designers a formal assessment of the security level of a given (leaking) device from the conceptual stage (without added countermeasures) but also quantitatively estimates the amount of noise (increased by addition of externally added countermeasures or by tweaks in the design) to mitigate the ease of a DPA. This leads to computation of metrics such as the correlation coefficient, Measurements to Disclosure (MTD) (the number of traces to disclosure) or the information-theoretically sound metric, mutual information (MI), to quantitatively estimate the security of a device at any given stage of the design. Some of our concrete observations are that longer transient runs (higher `FMAX`), which add more noise, come at the expense of increased simulation run times (high CPU time) which does not track back to significant changes in our security metrics. The complexity would further increase with the number of gates in case of a full cryptographic implementation such as the AES or the PRESENT block cipher.

We conclude that intrinsic physical noise in MOSFETs and its impact on cryptographic implementations has been thoroughly studied and it is expected to increase significantly with technology scaling.

Furthermore, we have also provided a comparison between CMOS and DDSLL logic based styles using transient noise analysis while providing results from several concrete first-order univariate security metrics. Our evaluations show,

1. that a noisy DDSLL logic style exhibits a lower SNR (and hence a lower MI) trend at different supply voltages compared to a noisy CMOS implementation, mainly due to the considerable increase in *intrinsic* physical noise between the two technologies.

2. We also show that the noise variances exhibit a strong independence w.r.t the inputs for a dual-rail logic implementation compared

to a CMOS design. While our results are for an 8-bit AES S-box, we believe similar trends will be observed for a full AES implementation.

3. The results can also be used to discuss the effectiveness of using the intrinsic physical noise of the MOSFETs and its behavior across different cryptographic implementations and different supply voltages.

4. The extension of this work to larger circuits (especially for dual rail styles) and correlated noise sources at the cost of a larger area overhead while maintaining a higher security level compared to CMOS remains an open and interesting question.

Chapter 6

Noise Analysis of a Measurement Setup

In this chapter we further investigate using our design methodology the impact and behavior of intrinsic noise on the security evaluation taking into consideration the internal capacitance, inductances and resistances across PCBs, dies and ground cable. Through this case study, we are able to evaluate the impact of interconnects on external observation-s/measurements and the impact on intrinsic noise allowing for a more holistic approach to(pre)-evaluation of crypto implementations through simulations.

6.1 Motivation

Design of (secure) cryptographic circuits beginning at the RTL involves conversion of the high level synthesizable VHDL code to a gate-level netlist, followed by layout of the implemented design. This involves the floor-planning stage, wherein the peripherals of the die on which the design will be mapped, are defined. These include the distribution of the I/O pads, placement of the power supply (single/multiple voltage supplies), GND rails, etc. These are the peripherals which will be interfacing with the external artefacts such as the PCB interfaces, the packaging and the sockets. After floor-planning, the design is now ready to be placed and routed. Following a successful place and route (P&R), the design is exported as GDSII file to the foundry, where it is now ready to be mapped on-to a physical die (i.e., on silicon) and production can begin. Post fabrication, the testing and evaluation of the dies begins once they have been received from the foundry. A

PCB measurement board is now ready to house-in the die and begin the process of collecting the measurements. Apart from measuring the electronic aspects of the implemented design, which includes parameters such as meeting performance and noise margins, the supply voltage ranges within which the device can operate, security evaluations are also carried by the cryptographic designers, potentially involving SCA (such as Differential Power Analysis (DPA) [63], Template Attacks (TAs) [24] or Correlation Power Analysis (CPA) [20]). Only at this stage, are the vulnerabilities of the complete design (notwithstanding the built-in countermeasures) exposed to the designers, which are often attributed to the measurement set-up employed by an adversary and have proven to be successful in breaking even the most (mathematically/cryptographically secure) implementations.

Although, post-synthesis and post-layouts simulations effectively cover the performance margins (and electronic aspects) of the die during design cycle, there is (yet) no clear defined security evaluation methodology which focuses on the security evaluations at the post-synthesis or the post-layout stage. Consequently, if a fabricated design is found to be broken, either due to an excellent measurement setup or equally by an average adversarial set-up, it entails additional cost and time-to-market for the design. Additional countermeasures, may be introduced to compensate (where possible), otherwise the fabricated die has to revert to the design board and the process repeated. This process is costly and delays the production cycle further. This necessitates the motivation to have a security evaluation methodology, based on simulations, which is robust enough to provide guidelines at the pre-layout stage as well allow for modifications of the design, depending upon the evaluated results.

6.2 Existing State-of-the-Art

Using simulations for side-channel leakage evaluations has been previously studied in [68] [69] [103], although these earlier works stressed on the evaluation of leakage implementations solely from a simulational point of view. Little consideration was given towards the modeling of any peripheral artefacts and its corresponding impact on the simulated leakage traces and their subsequent security evaluations. Nevertheless, they provide a strong motivation to simulate and evaluate the resistance of an implementation at design time. Building on this flow, importance was given to physical security and their corresponding optimizations in [101] [124].

One of the first works modeling equivalent circuit models for cryp-

tographic designs was reported by Iokibe et al. [50]. In their work, the authors propose an equivalent circuit mode which models both the Power Distribution Network (PDN) of a circuit which implements the AES-128 cryptographic algorithm with a high level modeling of the impedance and the board and package models. Their results, nevertheless showed the correlation between the simulated traces (through their model) in agreement with the actual power traces obtained as a result of measurements. Nevertheless, throughout their analysis, the authors freely use decoupling capacitances to obtain simulated power traces (and suppress the informative leakage), and after a number of iterations (with different de-cap values), obtain a close agreement with the real world measurement traces. Further extensions of related works were attributed to the modeling of the peripheral microcontroller circuity [51] and related supply pins [84]. However, it is noteworthy to point out, that in all these works, previous actual measurements were already made on a prototype working chip; the corresponding traces obtained from simulations served as a validation of the proposed model, rather than as a methodology to evaluate the resistance of the implementation pre-fabrication. Although this is the conventional way of modeling designs, (i.e., modeling from measurements), for evaluating (or rather estimating) the intrinsic side-channel resistance of a cryptographic implementation pre-fabrication, the reverse approach (i.e., modeling followed by measurements) could also be beneficial in quantitatively estimating the countermeasures required (to secure the implementation) before the design goes to the foundry[1].

Kamel et al. [54] reported a strong correlation of the simulated power traces with actual power traces obtained post measurements, for a side-channel security evaluation use-case. In [54], the authors analyzed the impact of the PCB packaging, socket and the cable inductances, in sufficient detail, for a dual rail logic style implemented using full custom technology for a commercial 65nm PDK, with previously obtained actual measurements from the same chip. Their work is able to accurately correlate the security evaluation metrics (in this case, using the Perceived Information (PI)) for similar noise levels, as reported by actual measurements. Nevertheless, the authors do not incorporate any intrinsic physical noise in their simulations, rather merely model the noise as an additional added parameter (using MATLAB simulations) to study the impact of the security evaluation metric (the PI in this case) w.r.t the noise. Nevertheless, the work provides useful insights into the impact of PCB artefacts on the security evaluation of a device and we use it as a building foundation for our proposed methodology. Further in [6], which

[1]However, we do mention here that within the scope of this work, we limit ourselves (only) to the simulations of implementations taking IPN into account

model the PCB packaging effects, parasitics and oscilloscope probes are reported. The authors in [6] use a lumped RLC model using scattering parameters obtained as a results of 3D simulations. Although computationally sound, transient noise, coming intrinsically from the physical devices is not considered. Similarly, authors' in [13] use low-level digital simulations to evaluate the security of implementations at different stages of the digital design flow. Recent works reporting a renewed interest in simulation based methodologies evaluating the side channel resistance at netlist level include, but not limited to, [108, 13, 16, 109, 138]. In [109], for e.g., authors propose a framework for an automated analysis of implementations targeting design time; however, intrinsic noise is not accounted for. Noiseless simulations, as such, although capture the worst-case adversarial setup a design may encounter, but offer limited scope in the evaluation and understanding of physical noise which could also be beneficial for an implementation. Similarly in [108], although transient simulations have been performed, transient noise has not been accounted for which would lead to further correlations, both between the circuit and passive elements and within the noise sources themselves. Further works consolidating the role of simulations and electronic design automation, both from a bottom-up and system level perspective, can be found in [61] in which authors' highlight the importance of methodologies for e.g., Design-for-Security (DfS), as aforementioned.

6.3 Our Contributions

Transient noise analysis, which has already been shown as a successful methodology in evaluating the intrinsic noise of a design implementation earlier in the development phase [86] [87] [89], can be used in the modeling of peripheral artefacts which a design would encounter later in its evaluation and testing cycle. Although accurate modeling of such artefacts remains an open ended topic, especially in circuit design research, nevertheless, it offers cryptographic designers a quick and handy way of evaluating the figures-of-merit based on some easily available modeling scenarios in presence of intrinsic noise, thanks to our earlier introduced methodology.

In this chapter, we provide a first hand evaluation and modeling of artefacts which (possibly) impact the overall design of an implementation on a fully-working die; namely, we assess the impact of package, socket, the PCB board itself (along-with differential probes) and the V_{DD} and GND cable inductances which play a crucial role, both in the evaluation of a cryptographic setup in presence of intrinsic physical noise.

Our contributions are as follows:

- Using transient noise simulations, earlier in the development cycle, with modeling of the peripheral artefacts, can be used to analyze the intrinsic security of such implementations once they have been packaged and socketed.
- Additionally, the impact of the differential probes (or other sophisticated probes) and the cable inductances can be used to assess the impact of adversarial setup very earlier in the design development cycle.
- We analyze in detail, the effect of the supply voltage V_{DD} and the effect of the supply voltage on the PI metrics for the different cases implemented.
- We also present first hand evaluation results of the leakage function and distribution plots to back-up our claims.

6.4 Transient Noise Analysis of Measurement-based Artefacts

In this section we build on the existing methodology introduced in the preceding chapter to evaluate the security of a cryptographic design using intrinsic physical noise, i.e., the noise originating directly from the MOSFET devices in presence of measurement based artefacts. The various sources of noise, which a given MOSFET (or transistors) would be subjected to, have already been briefly covered in Section 2.3. We briefly remind the reader of the simulation methodology introduced in chapter 4. Building on the works from chapter 4 (and chapters **??** and 5), we implement the TIMESPAN (ref. chapter 4) methodology to further evaluate the addition of PCB artefacts modeling such as packaging effects, sockets, and the cable inductances. The validation of the aforementioned methodology consists of validating the claims (quantified by the PSD) from a "physics-based" perspective, and comparing it with the "security" evaluation metrics (such as SNR/PI) in presence of the measurement-based artefacts set-up. This allows for total validation of the evaluated results when applying the security evaluation metrics such that the noise values obtained can indeed be traced back to their electronic counterparts. This provides a sound validation to estimate (and evaluate) the side-channel resistance of a design (or early)-stage implementation (or netlist) exploiting the intrinsic physical noise using transient noise simulations.

6.4.1 Artefacts based Model

The estimation of first-hand security metrics, in a setting involving the modeling of measurement-based artefacts involve the (accurate) modeling of the Power-Distribution Network and the packaging and PCB effects, along with the necessary impedances, an active implementation might "see"; a very high level overview is shown in Figure 6.1.

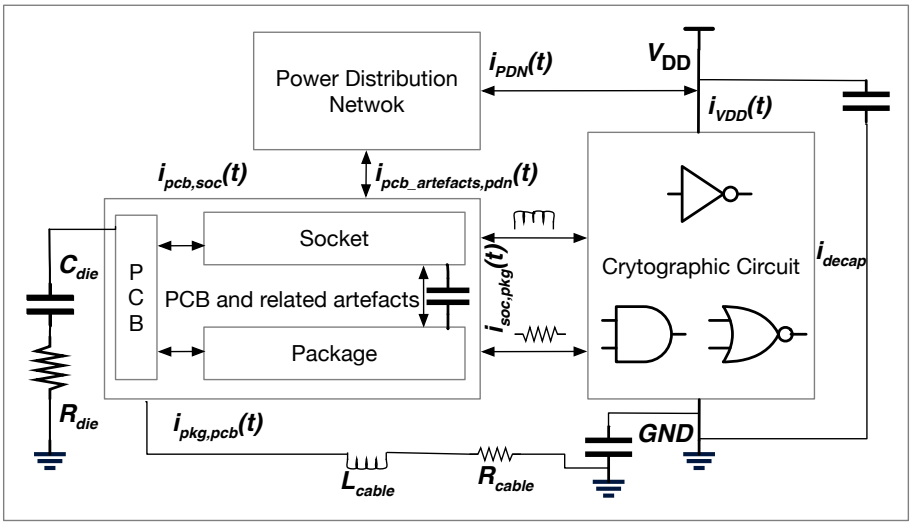


Fig. 6.1: High-Level Overview of the PCB artefacts interfacing the Cryptographic Circuit of interest.

As shown in Figure 6.1, the current $i_{V_{DD}}(t)$(which is a time-varying current measured at the supply voltage), V_{DD}, is not a constant current; but rather, is the algebraic sum of multiple currents correlated from different sources. We detail this in the following paragraphs.

A note on the PVT variation of the Devices We note that for the purposes, of this case study, we typically evaluate the subsequent mentioned cases solely in the presence of IPN and do not consider the impact caused due to the Process, Voltage and Temperature (PVT) variations (which could potentially also be seen as an external source of noise, rather than intrinsic). Intuitively, this allows us to consider the above mentioned cases as a worst case evaluation setup (scenario) wherein the only source of intrinsic side channel resistance present is offered only by the IPN and without *any* externally added circuit, mathematical or PVT based countermeasures. Nevertheless, we acknowledge the presence of such PVT-based variations (which, for instance, could we well simulated by Monte-Carlo simulations to depict the intra- or inter

die-to-die (D2D) or wafer-to-wafer (W2W) variations) as an interesting scope for further research and evaluation. As such, all devices in the context of this chapter (and in general, the thesis) have been simulated "as-such" as they are on the same die.

Supply Current The supply current, in case of evaluating circuits running cryptographic implementations, is the current measured at the supply voltage, i.e., at V_{DD}. For an ideal setup (which is the case of simulations conducted without transient noise),

$$i_{V_{DD}}(t) = i_{V_{DD}} \tag{6.1}$$

In case of simulations, which have noise added *externally*, the simulated current would be,

$$i_{V_{DD},sim}(t) = i_{V_{DD}} + i_{\sigma^2_{external}}, \tag{6.2}$$

where $\sigma^2_{external}$ is the noise added externally (most commonly, through MATLAB simulations). This added noise, often, is in close agreement with the actual noise measured from pre-existing measurement setups, and consequently, provides no significant insight into the physical noise, coming intrinsically from the implementation. Thanks to transient noise analysis methodology, TIMESPAN, introduced in this work, intrinsic noise can be simulated with/without the addition of any externally amplified noise, i.e.,

$$i_{V_{DD},sim}(t) = i_{V_{DD},\sigma^2_{intrinsic}}(t) + i_{\sigma^2_{external}}, \tag{6.3}$$

For the rest of the work, we consider the case, $\sigma^2_{external} = 0$, i.e., we do not consider the addition of any externally added noise to our simulation setup.

Leakage Current Leakage current, in a side-channel cryptographic analysis setting, differs from the "traditional" definition of leakage current encountered in electronic circuit design. Cryptographically speaking, the current (often referred to as “power” traces in side-channel literature), refers to the measured (or simulated) current at the supply voltage of the cryptographic circuit-of-interest (CIT). Often the CIT has a separate supply voltage rail, to maximize decoupling from other voltage supplies of the circuit (in order to minimize the cross talk and interference). Mathematically, this is can be represented as

$$\mathrm{L}(t) = \left\{ i_{1_{V_{DD},sim}}(t), i_{2_{V_{DD},sim}}(t), \ldots, i_{n-1_{V_{DD},sim}}t, i_{n_{V_{DD},sim}}(t) \right\} + \sigma^2_{total} \tag{6.4}$$

The "power" trace, is thus, a set of current values, for a given time t, which can then be converted to suitable Hamming weight values for further analysis. The noise, σ^2_{total}, is often considered as noise which can be filtered by the adversary (by repeated averaging over multiple such leakage traces) to obtain leakages on which further statistical tools can be applied.

However, the above equation (ref. Equation 6.4) would change when considering intrinsic noise. This is due to the "intrinsic" behavior of the noise which manifests as part of a sample current trace i(t), i.e., substituting for Equation 6.3 in Equation 6.4,

$$\mathrm{L}(t) = \left\{ i_{1_{V_{DD},\sigma^2_{1,intrinsic}}}(t), i_{2_{V_{DD},\sigma^2_{2,intrinsic}}}(t), \ldots, i_{n_{V_{DD},\sigma^2_{n,intrinsic}}}(t) \right\} \tag{6.5}$$

Splitting further,

$$\begin{aligned} \mathrm{L}(t) = \Big\{ & i_{1_{V_{DD}}}(t) + i_{1,\sigma^2_{1,intrinsic}}(t), \\ & i_{2_{V_{DD}}}(t) + i_{1,2,\sigma^2_{1,2,intrinsic}}(t) + i_{2,\sigma^2_{2,intrinsic}}(t), \ldots, \\ & i_{n_{V_{DD}}}(t) + i_{n-1,n\sigma^2_{n-1,n,intrinsic}}(t) + i_{n,\sigma^2_{n,intrinsic}}(t) \Big\} \end{aligned} \tag{6.6}$$

Finally, summing up, the total mathematical leakage obtained as a result of all the sampled points $t = \left\{ t_1, t_2, \ldots, t_n \right\}$ can be represented as

$$L(t) = \left\{ \mathrm{L}_{\mathrm{t}_1}, \mathrm{L}_{\mathrm{t}_2}, \ldots, \mathrm{L}_{\mathrm{t}_{n-1}}, \mathrm{L}_{\mathrm{t}_n} \right\} \tag{6.7}$$

Equation 6.6, although seemingly complex, is able to capture the correlation between the multiple noise sources which are present in the measurement and now, the "realistic" simulation of a cryptographic CIT. Although we limit ourselves to correlation degree two, i.e., we represent the correlation between current traces e.g., between $i_1(t)$ and $i_2(t)$ and so on with their respective noise sources and the correlated noise sources (interactions) between them. In practice, the leakage current measured experimentally, will not be limited to a correlation between any two preceding traces; rather it will be the interaction and mathematical correlation of multiple sources, which would result in a stronger multivariate scenario (which is out of the scope of this work). The goal here is to show that these possible interactions of the multiple noise sources (again we do not yet define here what these noise sources could be; we visit this in the subsequent section), result in a rather complex correlation of the leakage current and the noise, which is difficult for any adversary with a decent measurement setup to filter out easily.

Noise Sources Intrinsic noise sources has been sufficiently covered in preceding chapters. The physical intrinsic noise coming from a transistor is attributed to the random fluctuations of the electrons within the substrate, among other factors. On a circuit design level, such random fluctuations between hundreds (or thousands, depending upon the design complexity) of transistors connected across a critical path in a cryptographic design, often leads to complex correlations between the voltages/currents of the interconnected transistors and subsequently, the noise powers of different transistors (which again is bias-dependent). In addition to this, the interactions of these noise sources, which have been shown to behave as intrinsic current sources themselves, with the residual capacitances and impedances results in further mixing of the intrinsic noise with circuit level noise. This noise, although random, is highly correlated with each of the elements present in its vicinity and subsequently affects the timing and performance of the design. Multiplying such correlations would ideally, increase the total amount of such noise, impacting the security evaluation of a device. This effect is manifested by changes in the shape of the normal distributions, which we detail further.

6.4.2 Artefacts based use-cases

To assess the impact of the peripheral and PCB artefacts on the intrinsic noise and its corresponding effect in analyzing the evaluation metrics, we need to carefully decouple each effect; i.e., each effect has to be gradually introduced into the simulations and assessed accordingly. Although, theoretically, a large number of such use-cases can be built (depending upon the degree of granularity we choose), nevertheless, not very significant improvements (or changes) are expected. For these reasons, we detail four cases, which we believe provide a good understanding and a gradual refinement of the simulation parameters, without including too many parameters at once, so that the analysis does not become cumbersome.

Artefacts based Model Rationale The rationale for the gradual inclusion of the artefacts, which includes both the Power Distribution Network (PDN) and the modeling of the auxiliary artefacts such as the packaging and sockets, the wirebonds, and the interconnects is motivated by Figure 6.2 which depicts a representation of the package and wire SPICE model. Extensions to this model covering the below relevant cases, for our case-study, are followed. Although exact values from manufacturers are not always available, the best effort has been made to obtain values as closely related to the type of packaging, sockets, and

V_{DD} and GND cable inductances employed in the study.

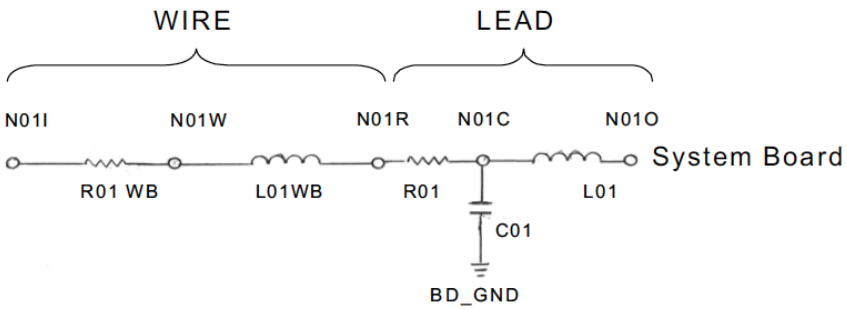


Fig. 6.2: Spice Model representing the PCB artefacts interfacing a Cryptographic Circuit of interest [1]

Table 6.1: Different Equivalent Circuit Models [54]

Model	Equivalent Circuit Model
1	1 kΩ
2	1 kΩ + differential probe + package + socket
3	1 kΩ + differential probe + package + socket + V_{DD} cable
4	1kΩ + differential probe + package + socket + V_{DD} cable + GND inductive cable

Case 1 Case 1 refers to the unprotected implementation being simulated *without* any PCB/peripheral effects. These correspond to the previous simulations settings detailed in Chapter 3. Design of the AES S-box is the same as previous simulations settings and the transistor siz-

ing has been chosen to be of the (minimum) standard-cells, to maximize the effect of the intrinsic noise (i.e., the flicker noise). Although in the previous simulations, we have chosen fully custom designed minimum sized transistor sizes, which work well, in a stand-alone setting without any peripheral effects, it leads to a deterioration of the performance (especially noise margins and timing delays) as soon as peripheral effects are included. To maintain coherency and for the purposes of brevity, in this case study, all the design of the chosen AES S-box have been designed in a semi-custom fashion using the minimum sized transistors (available as standard cells from the 28nm FDSOI PDK). As such, any changes in performance will now be coherent with the corresponding changes for all of the cases and do not contradict the previously mentioned claims. We refer to this case as the control case.

Case 2 Case 2 refers to the unprotected implementation of the AES S-box in a setting where the effects of packaging and sockets (along with differential probes) has been included. Individually, we accounted for each of the three sub-cases separately, i.e., package, sockets and differential probes, and did not see any significant changes in the results. Combining them together, however, and comparing with Case 1 provides useful insights into the design of a cryptographic implementation from an electronic perspective. Intuitively, the effects of package and socket should ideally be considered together as they form part of the broader packaging aspect.

Case 3 Case 3 refers to the unprotected implementation of the AES S-box in a setting wherein addition to Case 1 and Case 2, the effects of V_{DD} cable has also been included. The V_{DD} cable refers to the cable which connects the power supply to the CIT. Capacitances and inductances formed between this cable and the input/output signals (or pins) have been modeled and included in the simulation parameters. This effect is of particular importance in modeling for 2 main reasons,

- Firstly, it provides the relative modeling of the intrinsic capacitances and inductances with respect to each of the input and output pins; this needs to be included as each pin will "see" its corresponding capacitance and inductance, and

- secondly, the inclusion of such capacitances, inductance (and resistances) leads to the creation of an RLC-network whose effects are of interest in analyzing the cryptographic signal and noise. Additionally, we further investigate if such "stray" impedances lead to

an "Excessive" filtering of the signal and noise, as is conventionally believed.

Further more, we also analyze the impact on the security of an implementation due to the effect of the filtering. A change in the security level correlates either to the addition of noise or removal of noise, for a given (more or less) constant signal value.

Case 4 Case 4 refers to the final use-case of this analysis. In addition to taking into the effects encountered in Cases 1, 2 and 3 respectively, Case 4 includes the effect of the GND cable inductance. The GND cable is, in most of the cases, the longest cable of an experimental setup. It could be argued the same for the power supply cable, V_{DD}, but as it happens in most cryptographic (or rather side-channel measurement setups), multiple power supplies are usually used. Auxiliary circuit is normally supplied power by voltages supplies which have been carefully decoupled with the power supply unit of the cryptographic CIT. This decoupling of power supplies is beneficial for principally two reasons,

- Firstly, the power (or rather current) being measured for the crypto CIT, should be "true" reflection of the sum (or mathematical aggregate) of the switching currents (from the transistors) which are present only inside the cryptographic implementation and influence the crypto metrics. Switching currents encountered from auxiliary/additional logic could impact both the performance and measurement (or rather simulational observation) of such parameters.

- Secondly, any activity that uses other supplies will be considered as noise. So, shutting them down in order to avoid having additional noise coming from other circuits in order to evaluate the intrinsic noise is what needs to be done.

Furthermore, as described in previous paragraphs, the noise being injected from the additional logic (or supplies, are possible sources of algorithmic noise) would simply lead to a false sense of security, due to the effect of increased noise current. Consequently, assessing the impact of the intrinsic noise coming "only" from the transistors of the implementation would not be correctly accounted for; hence the need to decouple the power supplies.

In addition to the aforementioned conclusion sometimes during side-channel measurements the GND pin is also used (instead of the V_{DD} pin) to monitor the side-channel current. Analyzing the effect of GND

(cable) inductances could provide useful insights in the evaluation and early assessment of pre-layout designs.

6.4.3 Simulation setup

The experimental (or rather the simulational setup) consists of the same settings (as already described in chapters 3, 4 and 5 to maintain consistency) and the circuit models (described in section 6.4) used for the analyzing the impact of PCB artefacts on a cryptographic implementation. Since the analysis extends to the AES S-Box only (of course, a full AES can also be considered, but the S-box is in itself a highly active area of research still, both for cryptographers and for hardware designers, partly because of the non-linearities it offers and partly, due to the various implementations which it can be tuned into, each having its own separate FoMs). We implement the setup as per the high-level overview shown in Figure 6.1.

6.5 Experimental Results and Analysis

In this section, we provide experimental validation of our introduced framework. We first present the transient noise simulations results using the PSD, for two different supply voltages, which quantify the noise, from an "electronic" perspective. Furthermore, thanks to the intrinsic noise simulations, we are able to simulate noisy traces of the cryptographic implementation under test (in this case an 8-bit AES S-box), and perform side-channel assessment using the security evaluation metrics such as SNR and the PI and TVLA already aforementioned. We finally, correlate the values between the two approaches as a validation of our framework.

Pre-layout simulations are considered; the reason for this is two fold,:

- Firstly, pre-layout simulations allows us to capture the worst-case (or the closest "true-case") possible scenario where-in the value of the signal will be expected, the most, of course naturally (due to the presence of *only* intrinsic noise and no other countermeasure).
- Changes, to the design can be made much more easily at the pre-layout level compared to the post-layout stage where already considerable design effort has already been made while making the layout (and doing the post-layout extraction which is expected to exacerbate the noise due the presence of additional parasitics).

To re-iterate, the goal of this study is to analyze the impact of the intrinsic noise only due to the peripheral artefacts; additional parasitics

such as the ones from the post-layout simulations are out-of the scope of this study and have been purposefully not included for the reasons cited above. However, we believe similar conclusions can indeed by applied to the same,partly because they would produce the same effects, impacting both the performance and security evaluation metrics, however, in an exaggerated way. We leave this as an area of further exploratory research.

Analyzing cryptographic implementations for side-channel security evaluations using circuit (or SPICE) simulations remains a niche area of academic research and we urge the readers for further exploration in this field. We assert that with the right tools (i.e., a correct modeling of the circuit and the experimental setup backed up by the already strong mathematical proofs) would go a long way in analyzing the security of such implementations at pre-layout stage. On the other hand, strong mathematically-backed techniques such as using Domain-Oriented Masking (DOM) [39] or Threshold Implementations (TI) [90] are often broken [67] (i.e., the key successfully recovered) with traces much lower than the postulated theoretical limit.

6.5.1 Transient Noise Analysis Results

In this subsection, we analyze the PSD curves of the different cases and compare them to the estimated cryptographic FoMs; i.e., we first present the PSDs for all the analyzed cases (for both 0.5V and 0.9V) and then refer to the values obtained from a cryptographic analysis.

Simulation Parameters We detail here the simulation parameters for the subsequent analyses carried out in this chapter. By default, these are the conditions of operation unless specifically captioned otherwise (for instance, a different V_{DD}).

Inputs $\in$ {0,1,2, ..., 254,255}, $\mathrm{N}_{traces} = 40$, T = 51.2 μs and dt = 40 ps for a total budget

$$N_{traces} \cdot \frac{T}{dt} \tag{6.8}$$

of $\sim 512 \times 10^6$ for a given supply V_{DD}; `FMIN` = 1/T and `FMAX` = (1/2)$*$dt (=12.5 GHz) are chosen as per the described methodology.

Estimation of the Noise PSD To quantify the noise present in an un-protected implementation of the AES Sub Bytes (Sbox) and to assess the impact of the gradual addition of the measurement artefacts, we use the Power Spectral Density method. The PSD plots the noise power over a range of the working frequencies and gives an accurate picture of the behavior of the noise (or the signal) and its value at each

corresponding frequency. Consequently, this is a very useful tool for cryptographic designers who are interested in achieving high-throughput or performance intensive digital logic designs with often neglected or limited knowledge of the corresponding higher order frequency effects the design will induce during operation. The PSD allows us the designer to estimate the noise (intrinsic noise) in this case to be quantified, building upon which additional countermeasures (at the circuit or gate or algorithmic level) can then be added.

To plot the noise PSD, we separate the noise (traces) from the transient-noise simulated traces (which contains both the signal and noise components) as shown in for Case 2 ref. Figure 6.3. The transient noise simulations are computed over the complete set of dynamic switching inputs for an AES-Sbox i.e., from 0 to 255, as in the preceding chapters when simulating transient noisy runs. The corresponding (averaged) PSD is then computed over these noise traces as shown in ref. Figure 6.4 for each of the respective cases (i.e., Cases 1, 2, 3 and 4).

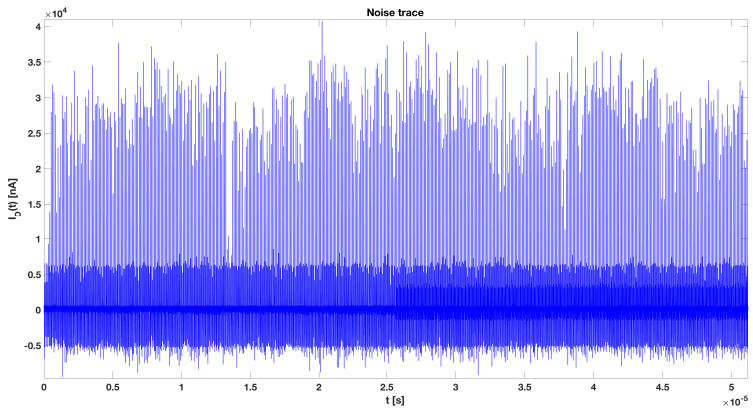


Fig. 6.3: Noise trace shown clearly segregated from the Transient-noise simulated trace for sample case 2 corresponding to V_{DD}=0.5V

For this case study, Figure 6.4 and Figure 6.5 depict the PSD of an AES Sbox with gradual addition of cryptographic artefacts, as per the methodology introduced in the preceding section. The plotted PSD is represented for all the possible current traces on which the design has been simulated upon, with the addition of intrinsic physical noise. Compared to other works in literature, which often introduce an arbitrary noise parameter (either through MATLAB additions or simply don't take noise into consideration at all), this work, for the first time to the best of our knowledge, incorporates actual physical noise from the transistors

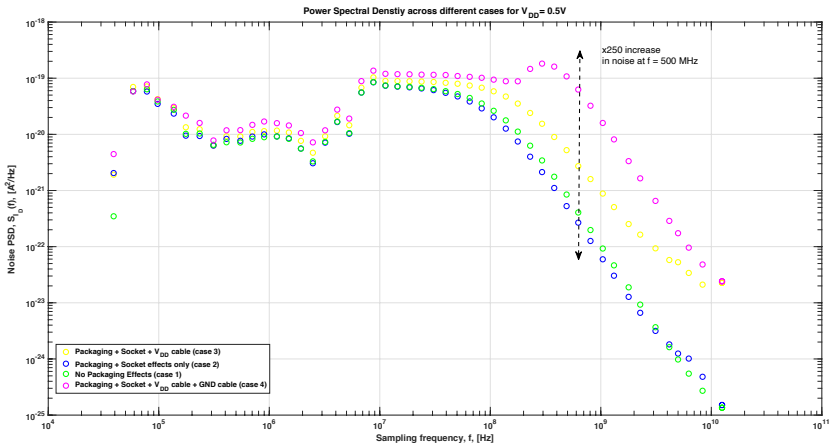


Fig. 6.4: PSD across different cases for V_{DD}=0.5V for transient noisy runs and the values computed for the POI

and assesses the impact on the security evaluations FoMs.

PSD at Lower Supply Voltage Figure 6.4 presents the PSD for the implementation at a supply voltage, $V_{\mathrm{DD}} = 0.5$V. Starting with the lowest value of noise (for a given frequency of say, 500MHz), the green curve clearly shows the lowest noise PSD. With the addition of artefacts, moving on from the addition of package and socket models, to addition of introduced V_{DD} cables and finally the GND cable, we see an increase of more than two orders of magnitude between the control case (i.e., Case 1) and the case which includes all the models simulating an actual measurement setup (i.e., Case 4).

PSD at Higher Supply Voltage Figure 6.5 shows the PSD plots for the implementation at a higher supply voltage, (closer to the nominal or operating voltage), V_{DD}=0.9V. Here the trend is again similar to the aforementioned lower voltage case; transient noise, in this case, is also more than two orders of magnitude higher compared to the control case.

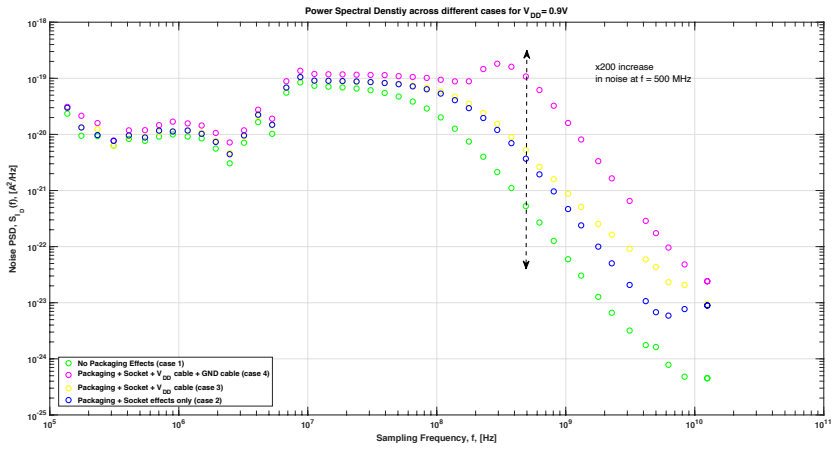


Fig. 6.5: PSD across different cases for V_{DD}=0.9V

Effect of the Impedance Filtering Figure 6.4 and Figure 6.5 also shows the roll-off of the information carrying components at higher frequencies saturating with the increase of the artefacts (due the RC-filtering effect). This is of interest in the assessment of side-channel leakage evaluations, which often are done at a higher mathematical level (and often neglecting the impact of the operating frequency while performing such evaluations). Figure 6.4 clearly shows that a higher noise level can be achieved at higher operating frequencies (for Cases 3 and 4 say ≈500 MHz), which clearly show the impact of the measurement artefacts on the IPN is more visible at higher frequencies. However, we note here the following, that the goal is to show the PSD trends (rather than actual numerical values), computed through transient noisy runs, which is memory (data complexity) and computationally (time-complexity) intensive. Subsequently, simulations (using transient noise for PSD analysis) beginning from f= 0 (or 1) Hz are almost always never done (in case of such larger implementations) and hence computing the total noise from the PSD using transient noisy runs would ideally, not provide the complete picture; nevertheless, such analyzes does provide a qualitative appreciation of the relative trends.

To further validate our approach from a cryptographic perspective, we now present the results of our security evaluations for the aforementioned cases, considering a univariate leakage model.

6.5.2 Security Evaluations

Assessing the impact of transient noise on the physical security of cryptographic implementations remains an open question. Building on the works from Chapter 4, this case study extends the introduced methodology to the case of assessing the impact of measurement artefacts and their modeling for extraction of meaningful security FoMs, which could be extended to a side-channel analysis practical evaluation/attack scenario. However, notably, analog SPICE simulations often take a huge simulation time and the complexity grows with the increased circuit sizes and inclusion of more artefacts parameters and models; not to mention, that using post-layout simulations with the aforementioned constraints would simply result in an exponential increase of the simulation time and the utilization of the computing resources. For these reasons, and as mentioned in section, we present results for the pre-layout based simulations and nevertheless conclude, that similar trends will be extrapolated to the post-layout simulation cases, which remains an interesting and open research question.

The Case of SNR: Classical Univariate

SNR based security evaluations provides the most basic security analysis in the evaluation of a cryptographic implementation. Although, the leakages encountered in practical scenarios are often multivariate in nature and increasingly complex (additionally, if masking countermeasures are applied), the SNR remains a classical metric to provide a first-hand assessment of the leakage (especially at pre-layout design levels where measurement and collection of huge number of traces for side-channel analysis is out of the question).

Multiple noisy transient runs (simulated using the Eldo .noisetran command) is able to capture the effect of measuring leakages across multiple measurements with noise included, which can then be further processed to compute security metrics, similar to the case of post-processing employed after side-channel based measurements. We plot the transient noisy runs across all the switching inputs (i.e., from 0 to 255 in case of a 8-bit AESSbox) across multiple noisy runs. For representational purpose, we only depict transient noisy leakages for a single random noisy run. Figure 6.6 and Figure 6.7 shows the leakage traces for the control case (i.e., Case 1) and the most complex case (i.e., Case 4 which includes all

the artefacts models) for a single noisy run for all the possible data input cases. Such simulations provide more realistic views of the leakages, as is evident by the shape of the waveform and the data-dependence variation and the noise variation (not shown here but among the different noisy traces themselves for a given data input). It is this variation (both due to data-dependence and the noise variations encountered due to multiple runs), which allows for a side-channel security (pre)-evaluation and assessment.

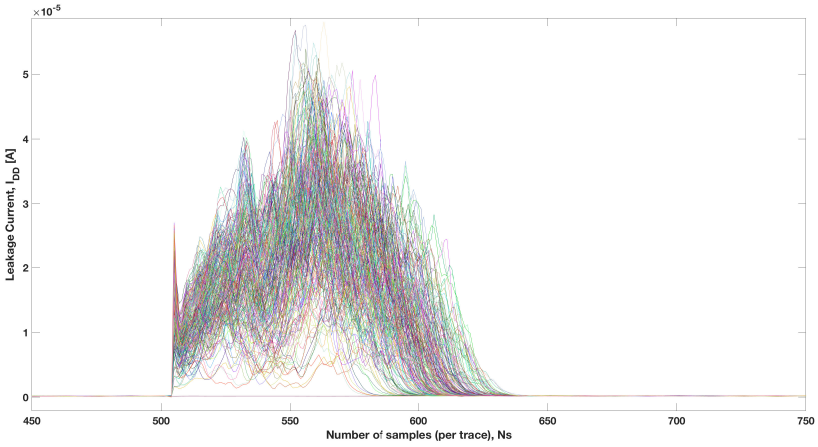


Fig. 6.6: Exemplary Transient Noisy Leakage traces for case 1 for V_{DD} =0.5V

Figure 6.6 (corresponding to Case 1) and Figure 6.7 (corresponding to Case 4) clearly shows the dynamic and static region of the current, with the black highlighted lines in ref. Figure 6.7 showing the possible area of interest in which the POI may be searched. The highlighted dynamic region shows the variation of the switching dynamic current. Compared to the control case (ref. Figure 6.6), a number of differences can be clearly enumerated.

1. The POI in Case 1 is in a much tighter region i.e., lesser number of samples are needed to compute (or search) for the POI than in the Case 4 where it is more widespread.

2. The impact and variation on the intrinsic noise is clearly visible not only in the dynamic region, but more so on the static region as well.

3. Overall, there is a decrease in the magnitude of the leakage current in Case 4 compared to Case 1. This is to be expected as the

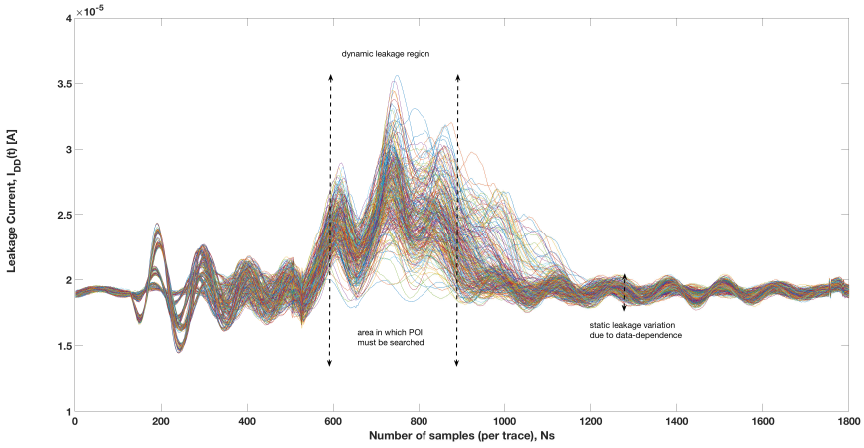


Fig. 6.7: Exemplary Transient Noisy Leakage traces for case 4 for V_{DD} =0.5V

interconnect network and the correlated noise form a complex filtering setup for Case 4 (ref. Figure 6.7) than in Case 1 (ref. Figure 6.6).

The variations in the current traces themselves (thanks to the data-dependency coupled with the noise effects both due to the intrinsic physical noise and the cable inductive effects) provide a strong motivation for further evaluations. Also, notably data-dependency on the static region is observed.

Transient Analysis of the SNR A transient[2] representation of the SNR allows for the visual inspection of the leakage points. Usually, the point of maximum SNR is chosen as the POI. An exemplary SNR trace is shown in Figure 6.8 for the case 1 (wherein none of the packaging effects have been considered) across different supply voltages. Multiple leakage points are shown and normalized SNR shown for clarity; however the maximum SNR chosen could be (actually) higher. We discuss the following points:

1. A quasi-linear dependence of the SNR with the supply voltage (between the regions with samples numbered 500-600) is observed in Figure 6.8, which is in agreement with previous such representations.

[2]although a transient representation is associated with a temporal values, we map the transient time values into equivalent sample representations

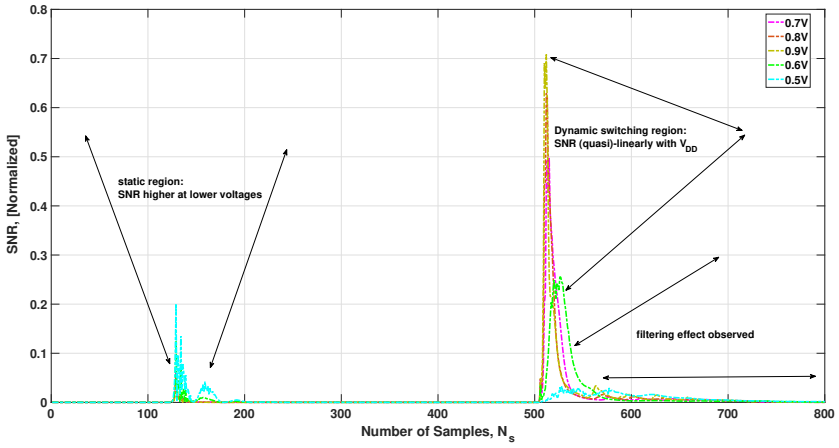


Fig. 6.8: Exemplary SNR of Transient Noisy Leakage traces for Case 1 across varying V_{DD}s

2. A gradual fall of the SNR slope is observed for lower voltage of operation (i.e., corresponding to 0.5V and 0.6V) whereas a much sharper slope (spike) is observed for the higher V_{DD}s. This could be due to the fact that at lower voltages (say at 0.5V or 0.6V), the delay is higher compared to at higher voltages, and hence the slower transition compared to much faster transitions (say at 0.8V or 0.9V). Note the particularly longer delay in the 0.6V (falling) transition.

We plot the PDF, for the Cases 1 (above) and 4 (below) as shown in Figure 6.9. The PDF represents for two randomly chosen input transition current traces at the chosen POI (in this case, the time sample corresponding to the maximum value of the SNR). The top plot for the Case 1 wherein clear Gaussian, with two distinct (i.e., no overlapping) of the means (and variance) is represented. Subsequently, attacking such a well-defined distribution is easier and well documented. Compared to the (sub)-figure below, which represents the PDF for Case 4, for the same (two randomly chosen) inputs, an overlapping of the means is observed due to the increased noise in Case 4, thanks to the comparatively more correlations between adjacent noise samples. Attacking such convoluted (or tightly spaced) distributions could indeed require additional resources from an adversarial point-of-view and may possibly increase the attack complexity as the adversary may need to collect further samples to understand the distributions more clearly. We exercise a cautionary note here in that such results necessarily do not

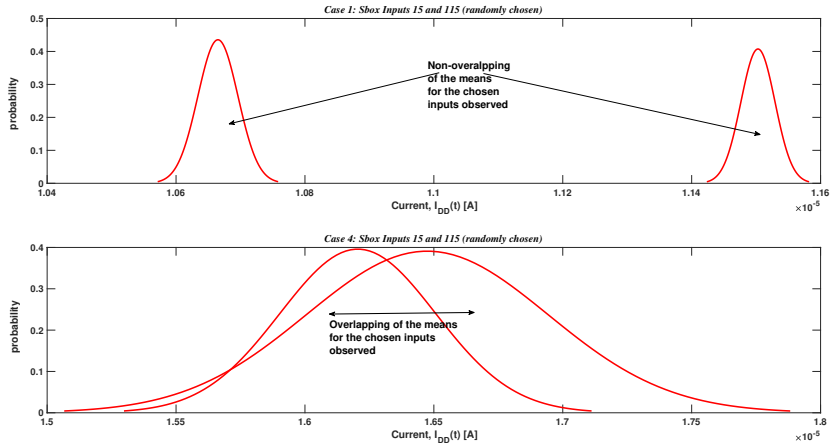


Fig. 6.9: PDF of maximum POI for case 1 (top) and case 4 (bottom) for 2 randomly chosen inputs across noisy transient runs for $V_{\mathrm{DD}} = 0.5\mathrm{V}$

indicate side-channel independence as the simulated conditions are for an un-protected implementation in presence of only IPN. With addition of countermeasures such as masking (or Threshold Implementations) in presence of IPN, we further expect the mixtures to be further convoluted and more tightly spaced, thus increasing the overall complexity of the attack.

6.5.3 Analysis using the Hamming Weight Leakage function

The HW model classically, models the leakage of a cryptographic implementation. However, for most of the systems in practice, the leakage function is unknown and consequently, the adversary requires a large number of traces in order to launch a recovery attack (e.g., using the Template Attacks based methodology). From an designer's perspective, lack of knowledge of the leakage function could lead to under-application (or even sometimes the over-application) of countermeasures, which would unnecessarily put unjust constraints on the designed specifications.

For the purposes of evaluating the leakages of such implementations, such as within the scope of this dissertation, and more specifically, this chapter, we analyze the leakage (starting with a Hamming Weight leakage model assumption) and plot the leakage values as a function of the Sbox inputs and the HW values. The goal here is *not* to accurately model the leakage function rather we hope to show that assuming some basic model, and incrementally building upon the cryptographic artefacts,

changes the inherent leakage model sufficiently, so as to take notice. Since the approach here is rather relative, i.e., by fixing the model for the control case, and evaluating subsequent changes, the choice of the original leakage model is rather irrelevant. We believe similar trends to be expected if an analysis would be performed using a different reference leakage model.

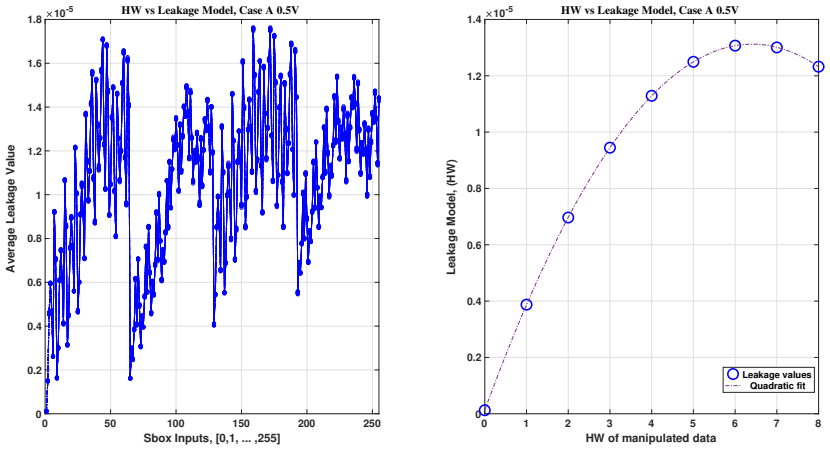


Fig. 6.10: Leakage Function of case 1 for $V_{\mathrm{DD}} = 0.5\mathrm{V}$

Leakage Model Analysis for Case 1 We make the following observations,

- The leakage function follows a highly repetitive pattern; this is partly due to the analysis over the full length of the input values, i.e., from 0 to 255.

- The leakage values seem de-correlated with each other; i.e., from the graph, no dependencies is observed.

- Moving to the (sub)-figure on the right, the leakage function is fitted for a quadratic function. This leads to the following interesting observations,

  Although the (sub)-figure on the left, hinted at the possibility of the leakage function being linear, the actual function could be possibly quadratic. What rather seems linear was the incremental behavior due to the behavior of the input values, which were being increased in a linear fashion. However, when analyzing on the basis

of HW values, clear observations on the quadratic nature of the function were indeed clearly observed.

It is remarkable that a very good fit was indeed observed for the leakage function, which indicates that thanks to the integration of the intrinsic noise with the current values, the quadratic nature is present. This conforms our earlier hypothesis that the intrinsic noise is rather a complex (multiplicative) function, than merely an additive function (compared to the AWGN/thermal noise).

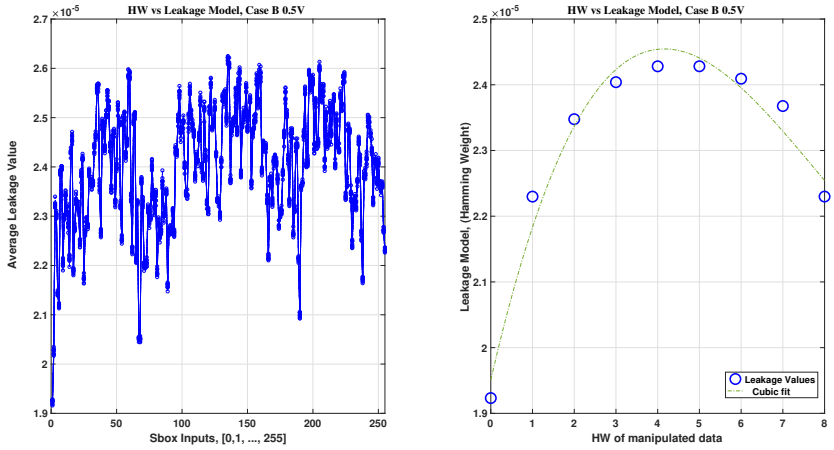


Fig. 6.11: Leakage Function of case 2 for $V_{\mathrm{DD}} = 0.5\mathrm{V}$

Leakage Model Analysis for Case 2 From Figure 6.11, we make the following observations,

1. We observe clear changes in the leakage function plotted w.r.t Sbox input values ((sub)-figure on the left). Comparing this to Figure 6.10, we can conclude the interactions of the transient noise with the PCB artefacts which have been introduced in this case model, leading to distortion in the conspicuous repetitive behavior as was observed before.

2. From the (sub)-figure on the right of Figure 6.11, the leakage function can no longer be fitted with a quadratic dependence. Rather, a best-fit now corresponds to a cubic fit, which indicates further possible interactions between the artefacts, the noise current and the observed current (leakage) values.

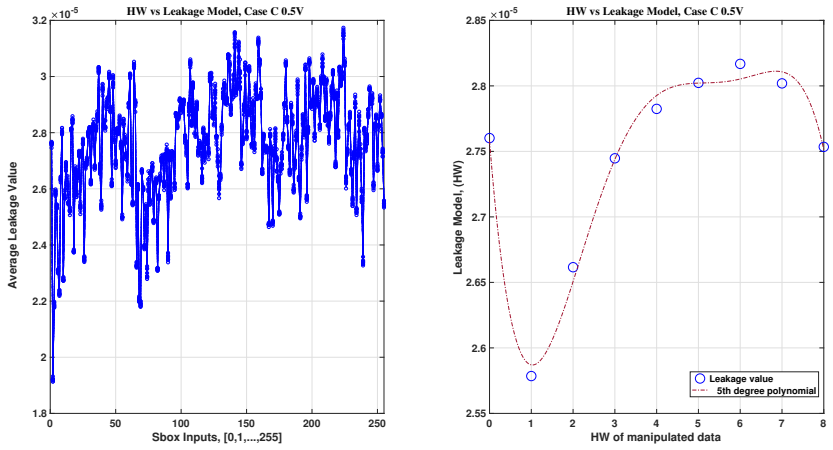


Fig. 6.12: Leakage Function of case 3 for $V_{\mathrm{DD}} = 0.5\mathrm{V}$

Leakage Model Analysis for Case 3 From Figure 6.12, we make the following observations,

1. The leakage function, for case 3, is further distorted thanks to the introduction of further dependent parameters and their possible correlations with the intrinsic transient noise. Although there is a repetitive pattern observable, the distribution of the samples is far less uniform as compared to the aforementioned previous two cases.

2. From the (sub)-figure on the right of Figure 6.12, the leakage function has now moved to a complex polynomial fitting equation with increased correlations between the various parameters of interest. Thanks to such interactions, the effect of the noise current is rather exaggerated, which as we see in the subsequent section, aids in the reduction of the information leakage.

Leakage Model Analysis for Case 4 From Figure 6.13, we make the following observations,

1. The leakage function, for case 4, can no longer be considered as having any observable repetitiveness. This is rather beneficial from a side-channel countermeasure perspective. Having a more uniform distribution of the leakage values, leads to a higher success probability of the attack failures, and consequently, more number of traces need to be adopted by the adversary.

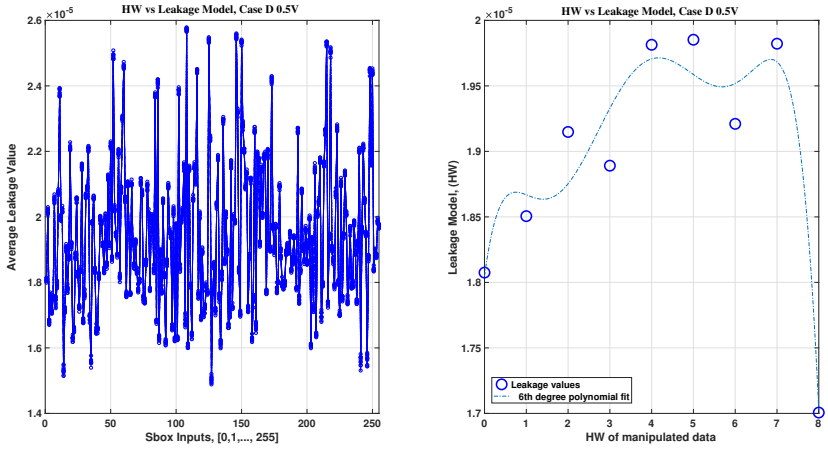


Fig. 6.13: Leakage Function of case 4 for $V_{\mathrm{DD}} = 0.5\mathrm{V}$

2. The leakage function has now adopted an extremely complex shape, far from its simple quadratic fit (as was observed from Figure 6.10). Such complex interactions, thanks to the exaggerated effect of the transient noise, could possibly offer substantial improvements in the design of cryptographic implementations when considering a side-channel attack scenario.

6.5.4 Analysis using the Noise Variance

As introduced in the preceding chapter, analysis using the computed noise variance values could also provide some qualitative insights into the behavior of the leakage function, and more so, on the noise levels expected from an implementation.

A Cautionary note on the Distributions and Analysis using Noise Variances Although the distributions represented below from Figures 6.14, 6.15, 6.16 and 6.17 have varying time scales, nevertheless, they provide a rudimentary (qualitative) appreciation of the correlation between the noise variance values; i.e., for instance, in Figure 6.14 corresponding to Case 1, the maximum value of the σ_2 is clearly almost an order of magnitude lower than for instance, say in Figure 6.17, corresponding to Case 4. Additionally, incremental changes in the maximum value of the noise variance can be observed from these distributions, with changes in their tightness, if not very quantitatively when moving from Case 1 to Case 4. We provide these distributions merely to complete our

different analyses and to familiarize the reader with an observable and pictorial trend when moving from Case 1 to Case 4.

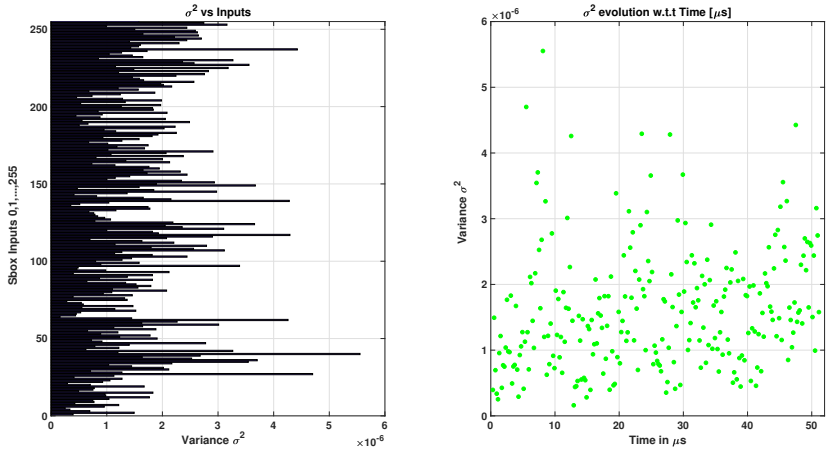


Fig. 6.14: Noise Variance for Case 1 as a function of the Inputs for a CMOS Sbox atV_{DD}=0.5V

Noise Variance Analysis for Case 1 From Figure 6.14, we make the following observations,

1. Non-uniform distribution of the noise variance values (represented by the bars on the left sub-figure) suggest that intrinsic noise, on its own, is not able to fully hide the effects of information leakage. This is suggested from a similar repetitive pattern being observed, when looked closely.

2. The same effect is manifested when plotting the noise variance values over time. Decreased correlation between the noisy time samples lead to a non-uniform distribution of the variances; thus easing the possibility of a side-channel attack.

Noise Variance Analysis for Case 2 From Figure 6.15, we make the following observations,

1. A tighter distribution of the noise variance samples is observed when compared to Figure 6.14, thanks to the increased correlation between the intrinsic noise and the introduced artefacts.

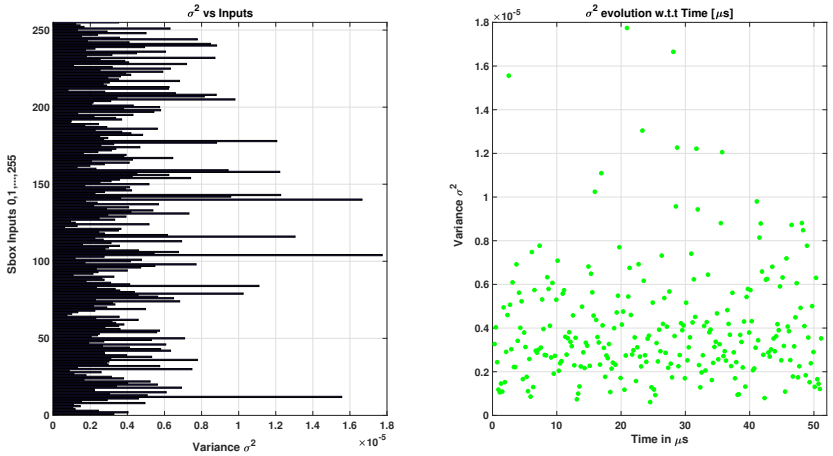


Fig. 6.15: Noise Variance for Case 2 as a function of the Inputs for a CMOS Sbox atV_{DD}=0.5V

2. The same effect is manifested when plotting the noise variance values over time. Increased correlation between the noisy time samples lead to a more uniform distribution of the variances.

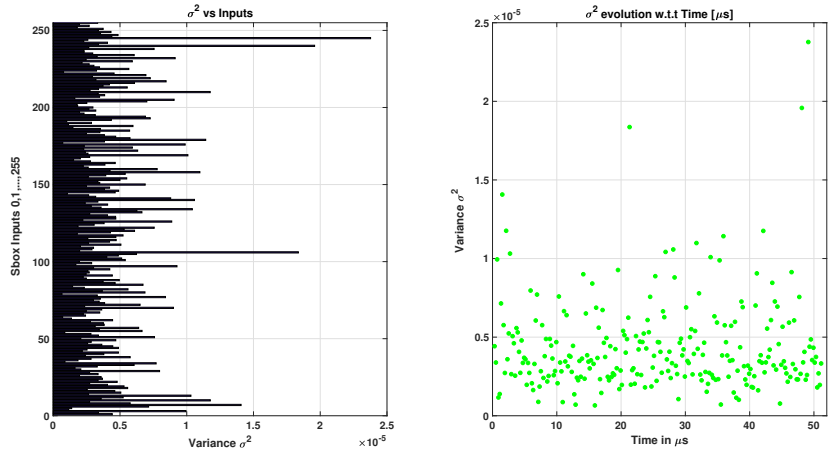


Fig. 6.16: Noise Variance for Case 3 as a function of the Inputs for a CMOS Sbox atV_{DD}=0.5V

Noise Variance Analysis for Case 3 From Figure 6.16, we make the following observations,

1. A tighter distribution of the noise variance samples is observed when compared to Figures 6.15 and 6.14, thanks to the increased correlation between the intrinsic noise and the introduced artefacts.

2. The same effect is manifested when plotting the noise variance values over time. Increased correlation between the noisy time samples lead to a more uniform distribution of the variances.

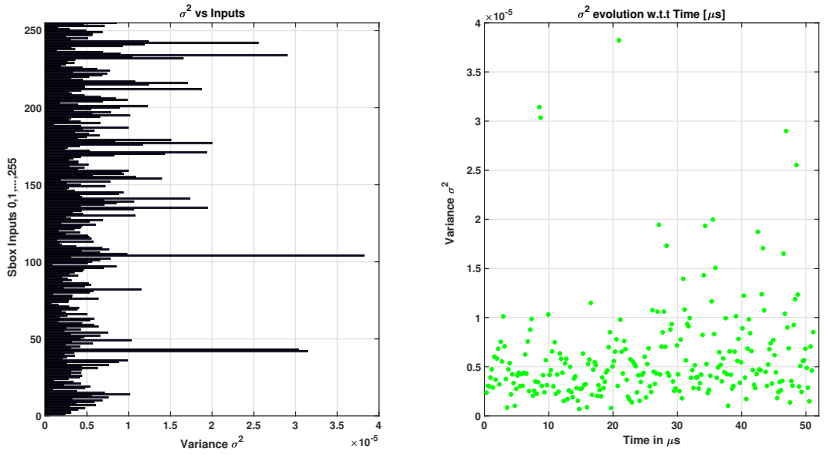


Fig. 6.17: Noise Variance for Case 4 as a function of the Inputs for a CMOS Sbox atV_{DD}=0.5V

Noise Variance Analysis for Case 4 From Figure 6.17, we make the following observations,

1. A much compact and tight distribution of the noise variance samples is observed when compared to the preceding figures. Thanks to the effect of the GNDcable inductance, the effect of the intrinsic physical noise is indeed exemplified.

2. The same effect is manifested when plotting the noise variance values over time. The noise variance values form a much compact distribution leading to the increased noise and hence, increased adversarial effort. Also note, the increased noise variance values when moving across Figures 6.14 to Figure 6.17.

6.5.5 Information leakage

The SNR, although a reliable metric, often does not convey the full extent of the information leaking from an implementation. As described

in the preceding section, It merely gives an estimate of the signal and noise values without quantifying the full extent of the perceived security of an implementation and the corresponding effect on the information leakage if the noise is varied across the system. The Mutual Information (MI) metric, quantifies the information leaked with respect to the noise present in the system and accurately models the behavior of the system, without the need for an accurate leakage model needed to describe it. This is of particularly important for this case study, as a large circuit such as the AES S-box cannot, for simply obvious reasons, be modeled by equations taking into full account the behavior of the circuit, the number of transistors in the critical path, the frequency of operation (which again changes the critical path and hence the number of transistors), the effect of artefacts, whether it's limited to inclusion of packaging effects, socket effects, differential probes or the inductive cable effects and the parasitics. This back annotation is cumbersome, and more importantly, not generic enough and has to be re-modeled for each new design implementation. Additionally, the explosion of the simulational and (computational) complexity also needs to be accounted.

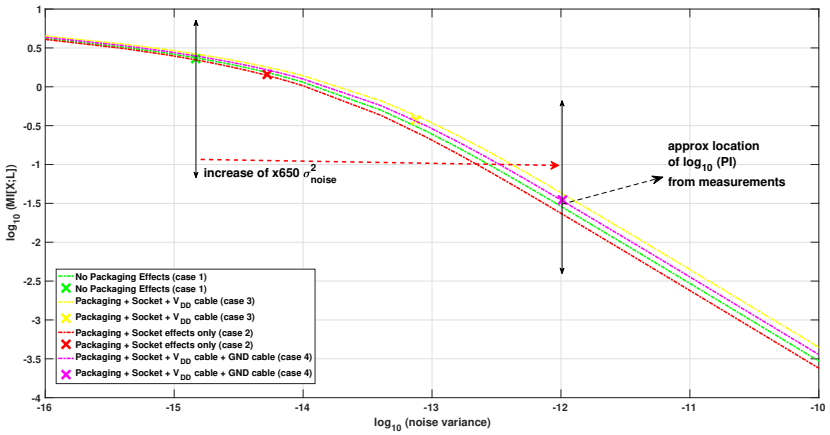


Fig. 6.18: MI curve across different cases for $V_{\mathrm{DD}} = 0.5\mathrm{V}$

MI at Lower Supply Voltage Figure 6.18 plots the MI at a lower supply voltage, V_{DD}=0.5V. A number of plots are shown on the curve in different colors, which highlight the different cases as described above. The following observations are made:

- The information leakage decreases, i.e., the MI decreases as the artefacts are added to an unprotected implementation. The green

curve shows the evolution of the MI for the control case. Additionally, the green marker indicates the amount of noise which was calculated, based on the chosen POI. To re-iterate, these values hold for a univariate assumption; further analysis is required for the computation of multi-variate covariances, which is out of the scope of this work and left as an open question.

- The magenta plots the evolution of the MI for case 4. Thanks to the increases correlations of the intrinsic noise with the artefacts, a good security level can be achieved, although at the cost of additional noise. The corresponding marker indicates the amount of noise which is required to obtain the security level. More than two-orders of magnitude of noise is observed between the control case and case 4.

A Cautionary Note on the Limitations of such Analyses From the preceding discussion, it may appear that a univariate analyses relaxes the conclusions of the MI considerably especially, in case of a measurement setup scenario wherein (active)[3] filtering was not employed. However, we exercise caution here that such (external) filtering mechanisms could indeed inject spurious noise and further aid in the reduction of the informative signal (leading to possibly better and lower MI values), their impact on the IPN may be of little or no considerable effect. However, in the evaluation of a multi-dimensional multivariate case, an external filtering mechanism could indeed allow for an easier analysis (through the tacit conversion of a multivariate case to a univariate case through, for instance, using a window function in analyzing the frequency response) and then the effect of IPN on each variation could indeed be separately analyzed and concluded. We leave the exploration of such an interesting case as an open question for further research.

MI at Higher Supply Voltage Figure 6.19 plots the MI at a higher supply voltage, V_{DD}=0.9V. A number of plots are shown on the curve in different colors, which highlight the different cases as described above. The following observations are made:

- Similar to the lower supply voltage case, the MI decreases with addition of PCB artefacts. However, the impact of the gradual noise addition (i.e., in moving from case 1 to case 2 and then to case 3) is comparatively less stronger. This is partly due to the increased

[3]by active, we mean here a dedicated noise filtering setup in addition to the filtering caused due to the introduction of the artefacts

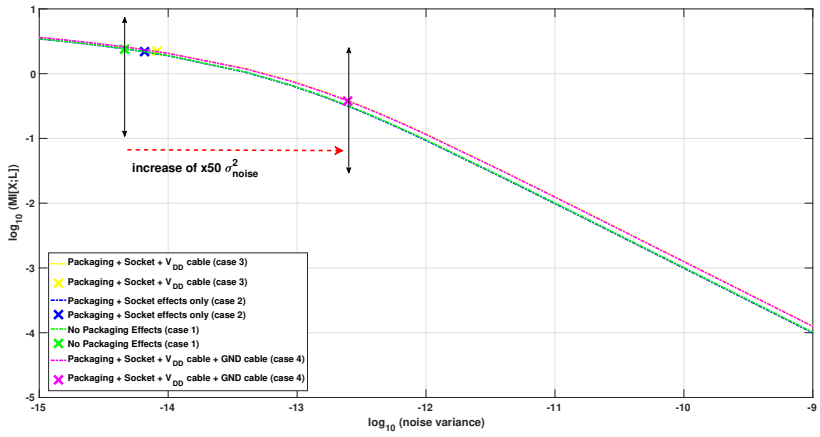


Fig. 6.19: MI curve across different cases for $V_{\mathrm{DD}} = 0.9\mathrm{V}$

voltage of operation, which typically compensates the correlations based induced delay, more pronounced at lower voltages.

- Considering transient noise, the impact on the security metric MI, is lesser when compared to the lower supply voltage case (i.e., lesser values of MI levels are reached for their corresponding noise values). Nevertheless, a direct extrapolation to the desired noise (or MI levels) is not recommended as a sane approach, since such analysis would not include the non-linear effects which may further come into play at higher noise levels.

6.6 Conclusions and Open Questions

In this chapter we extend our proposed methodology to evaluate and quantify the *intrinsic* MOSFET physical noise of cryptographic implementations using transient noise simulations to compute the first-order univariate security metrics using a pre-layout netlist in presence of physical artefacts such as PCB packaging, sockets and cable inductances encountered in real-life measurement setup scenarios. This allows us to discuss how physical noise sources can correlate with the already existing sources of noise (such as inductive bounces or capacitative effects of the differential probes) and alleviate the total noise, which could prove useful for cryptographic designers during early-design stages. A judicious choice of the model parameters, which would dictate the amount of noise which would intrinsically be present in the system, would allow designers (and

evaluators) to estimate the noise required at various levels of implementation, thus guaranteeing a minimum amount of noise, taking care not to compromise the performance margins (such as frequency of operation or the latency required). Although, this case study is relevant for the particular example of an AES S-box, we expect that it can nevertheless, be extended to any implementation using the provided methodology. Further improvements on the noise model as well as better modeling of an attack setup, extended to such as masking schemes which would research the impact of the intrinsic noise on such correlations, remains an interesting research question.

Chapter 7

Conclusions and Future Research

7.1 Thesis Contributions : A Summary

In this chapter, we briefly summarize the major contributions made in this dissertation.

Using the V_{DD} randomizer as a foundational case study which motivated the necessity of designing cryptographic implementations for the low-physical noise regions, this thesis first explored the validity of reducing the signal (using custom based logic styles which have been utilized to reduce the information bearing signal) in presence of technology scaling, and later thoroughly investigated detailed aspects of Intrinsic Physical Noise (IPN) for the design of such implementations.

The major contributions in terms of analyzing the impact of different logic styles with reference to technology scaling can be summarized as follows:

1. Investigated the validity of the signal reduction (as a countermeasure especially in the high-physical noise region) in the presence of technology scaling. Our analysis showed that dual-rail logic styles lose interest with technology scaling and their comparative advantage over CMOS-based styles decreases as we move from 65nm to 28nm nodes, both in terms of performance (reflected by the E_{op} metric) as well in security (quantified by the "Signal" metric).

2. CMOS based styles continue to be the design of choice on account of their ease of design, performance and security with respect to technology scaling and the design of such noisy CMOS based circuits as a choice for future works in the thesis.

The major contributions in terms of investigating the Intrinsic Physical Noise (IPN) in the context of this dissertation work are as follows:

1. Introduced a methodology built on a good understanding of simulation based tools, which is able to capture the transient noise behavior of designs and perform simulation-based assessments to accurately evaluate the noise related parameters of an implementation.

2. Validated the working of the methodology starting from a single device, to digital building blocks and finally to more complex designs. The impact of physical noise when moving to larger designs (i.e., how does the noise scale with the increase in the number of transistors) and also its dependence on supply voltage was assessed.

3. Validation of the results from a security (i.e., side-channel) perspective using state-of-the-art metrics such as the SNR and information theoretic metrics, including supply voltage dependence.

4. Proposed techniques to increase the noise in a simulation-based setting; to allow for a further assessment of the security levels which would be encountered at higher levels of implementation. Additionally, this also provides the designer an accurate estimation of the amount of noise, which needs to be injected into a system to guarantee some minimum security levels.

5. For early design-stage cryptographic security assessment, we discuss how it is possible to emulate with K_f the minimum level of noise, at which an adversary will not be able to further filter-out. Additionally, we discuss how to emulate lower noise levels from simulation to understand trends of noise reduction techniques.

6. As an additional validation of the scope of the introduced methodology, we re-visited the dual-rail based logic styles in presence of transient noise.

The validation of the introduced methodology in this thesis provided the motivation to further the understanding of Intrinsic Physical Noise in presence of a "measurement attack-based scenario" set-up. Keeping in mind, that (actual) measurements were outside the scope of this dissertation, nevertheless, it was necessary to understand the behavior of transient noise in such a (simulated) setting. To enumerate,

1. Proposed a high-level overview of a measurement based side-channel attack scenario taking transient noise into account.

2. Accurately evaluate the intrinsic physical noise, from both a physics perspective and from a cryptographic perspective.

3. Propose methods to increase the transient noise when assessing the impact due to set-up based artefacts; the role of such artefacts in increasing the correlations when transient noise is investigated and also leads the scope for further research.

7.2 Additional Research Contributions

In addition to the research conducted during the course of the PhD work on the Noise Analysis, some off-topics have also been a part of this work. To summarize them briefly,

Design and Implementation of the datapath of the GIMLI permutation in 28nm FDSOI technology as part of the submission to the NIST LWC competition. A fully-synthesizable RTL was formulated for the various rounds of the permutation. Additionally, serial and parallel implementations (with different rounds) were further synthesized. The design was implemented using the Cadence Genus Design tools and the results published as part of [12].

7.3 Future Research Directions

The thesis offered a highly interesting, interdisciplinary academic opportunity combining aspects from Semiconductor theory, Nanoelectronics and Applied cryptography in a Computer Aided Design setting. Keeping this in mind and the contributions made towards the finishing of this thesis, there are a number of areas into which multiple investigations can be pursued. Retaining the structure of the thesis' chapters, we present them in the same order.

Circuit-level countermeasures continue to be an area of interest in the broader (implementation-focused) academic research community as well as for commercial entities. Design of new Voltage Regulation Modules (such as the V_{DD} randomizer) continue to be of interest. As pointed out in the scope of this work, multiple voltage levels not only offer additional protection to a module from a circuit level perspective but also from the transient noise level. We further propose the investigation into the V_{BB} randomizer, taking full advantage of the flexibility (with respect to body biasing) and granularity (tunability of the body-biasing using both coarse and fine-grained techniques) 28nm technology offers. These two

coupled randomizers could indeed be a highly interesting case study in manipulating the noise levels.

With Moore's law approaching its limits and being replaced with Moore 2.0 or "Beyond Moore", the focus has shifted from the rapid shrinking of the technology node to a more architectural based design targeting the end-application. Keeping in mind, and noting that the semiconductor industry is accepting techniques such as FinFETs, 3D-stacking based architectures and novel devices based FETs (such as CNTFETs, MWFETs etc), will be a highly interesting case study to analyze from a cryptographic as well as understanding the behavior in intrinsic physical noise in such novel devices and architectures.

Within the scope of the presented research methodology, further investigations into speeding up the transient noise analyses, especially for larger cryptosystems. As described in the thesis, one of the major bottlenecks of this computer-aided design based case-studies was the explosion in the CPU runtime and the computational resources. Combining methods from the broader field of CAD theory could indeed offer more flexible and better performing simulation equipped analyses. Further more, the current set of transient analysis methods (including the ones introduced within this work) account for the Flicker and Thermal Noise. Including additional sources of noise such as the Random Telegraph Noise (RTN) could indeed be highly interesting research based topics. It is not necessary that the aforementioned sources of (newly suggested) noises be investigated (solely) in the context of side-channel analyses. For instance, typical cryptographic implementations implement serial and parallel modes of permutations; because of the difference in architectures, the noise distributions are expected to behave differently and worth investigating. Similarly, RTN, on account of its highly random behavior, could be utilized as a source of generating (random) numbers and could be implemented as a possible source of TRNG (True Random Number Generator).

Modeling of equivalent circuits, such as a side-channel measurement setup within the scope of this thesis, continues to be a continuing area of academic interest. A more accurate modeling of the noise behavior, the various sub-components associated (better closely matching packaging, socket and cable values) could indeed lead to further refinement of the suggested approach and better qualification towards the design time evaluation and assessment of side-channel countermeasures.

References

[1] application-notes.digichip.com. http://application-notes.digichip.com/314/314-66311.pdf. (Accessed on 11/11/2020).

[2] Learn more about fd-soi - stmicroelectronics. https://www.st.com/content/st_com/en/about/innovation---technology/FD-SOI/learn-more-about-fd-soi.html. (Accessed on 11/11/2020).

[3] Advanced encryption standard. In Eli Biham, editor, *Fast Software Encryption, 4th International Workshop, FSE '97, Haifa, Israel, January 20-22, 1997, Proceedings*, volume 1267 of *Lecture Notes in Computer Science*, pages 83–87. Springer, 1997.

[4] Parallel implementations of masking schemes and the bounded moment leakage model. Cryptology ePrint Archive, Report 2016/912, 2016. http://eprint.iacr.org/2016/912.

[5] Dakshi Agrawal, Bruce Archambeault, Josyula R Rao, and Pankaj Rohatgi. The em side—channel (s). In *International workshop on cryptographic hardware and embedded systems*, pages 29–45. Springer, 2002.

[6] M. Aigner, S. Mangard, F. Menichelli, R. Menicocci, M. Olivieri, T. Popp, G. Scotti, and A. Trifiletti. Side channel analysis resistant design flow. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4 pp.–2912, 2006.

[7] Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, pages 1–14, 2006.

[8] A. Asenov, A. R. Brown, J. H. Davies, S. Kaya, and G. Slavcheva. Simulation of intrinsic parameter fluctuations in decananometer and nanometer-scale MOSFETs. *IEEE Transactions on Electron Devices*, 50(9):1837–1852, Sept 2003.

[9] Michael Backes, Markus Dürmuth, Sebastian Gerling, Manfred Pinkal, and Caroline Sporleder. Acoustic side-channel attacks on printers. In *USENIX Security symposium*, pages 307–322, 2010.

[10] Karthik Baddam and Mark Zwolinski. Evaluation of dynamic voltage and frequency scaling as a differential power analysis countermeasure. In *20th International Conference on VLSI Design (VLSI Design 2007), Sixth International Conference on Embedded Systems (ICES 2007), 6-10 January 2007, Bangalore, India*, pages 854–862. IEEE Computer Society, 2007.

[11] D. Bellizia, S. Bongiovanni, P. MonsurrÒ, G. Scotti, and A. Trifiletti. Univariate power analysis attacks exploiting static dissipation of nanometer cmos vlsi circuits for cryptographic applications. *IEEE Transactions on Emerging Topics in Computing*, 5(3):329–339, 2017.

[12] Daniel J. Bernstein, Stefan Kölbl, Stefan Lucks, Pedro Maat Costa Massolino, Florian Mendel, Kashif Nawaz, Tobias Schneider, Peter Schwabe, François-Xavier Standaert, Yosuke Todo, and Benoît Viguier. Gimli : A cross-platform permutation. In *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, pages 299–320, 2017.

[13] Shivam Bhasin, Jean-Luc Danger, Tarik Graba, Yves Mathieu, Daisuke Fujimoto, and Makoto Nagata. Physical security evaluation at an early design-phase: A side-channel aware simulation methodology. In *Proceedings of International Workshop on Engineering Simulations for Cyber-Physical Systems*, ES4CPS '14, page 13–20, New York, NY, USA, 2013. Association for Computing Machinery.

[14] Bryan Black, Murali Annavaram, Ned Brekelbaum, John DeVale, Lei Jiang, Gabriel H Loh, Don McCaule, Pat Morrow, Donald W Nelson, Daniel Pantuso, et al. Die stacking (3d) microarchitecture. In *2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06)*, pages 469–479. IEEE, 2006.

[15] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[16] V Samadi Bokharaie and A Jahanian. Side-channel leakage assessment metrics and methodologies at design cycle: A case study for a cryptosystem. *Journal of Information Security and Applications*, 54:102561, 2020.

[17] David Bol, Dina Kamel, Denis Flandre, and Jean-Didier Legat. Nanometer MOSFET effects on the minimum-energy point of 45nm subthreshold logic. In *Proceedings of the 2009 International Symposium on Low Power Electronics and Design, 2009, San Fancisco, CA, USA, August 19-21, 2009*, pages 3–8, 2009.

[18] P. Bolcato and R. Poujois. A new approach for noise simulation in transient analysis. In *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, volume 2, pages 887–890 vol.2, 1992.

[19] Bardia Bozorgzadeh, Ehsan Zhian-Tabasy, and Ali Afzali-Kusha. Low-power high-performance logic style for low-voltage cmos technologies. In *2008 International Conference on Microelectronics*, pages 280–283. IEEE, 2008.

[20] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, pages 16–29, 2004.

[21] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises, 4th Edition.* Wiley, 2012.

[22] David Brumley and Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.

[23] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis

attacks. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 398–412, 1999.

[24] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, pages 13–28, 2002.

[25] Ba-Anh Dao, Trong-Thuc Hoang, Anh-Tien Le, Akira Tsukamoto, Kuniyasu Suzaki, and Cong-Kha Pham. Exploiting the back-gate biasing technique as a countermeasure against power analysis attacks. *IEEE Access*, 9:24768–24786, 2021.

[26] Santos Merino Del Pozo, François-Xavier Standaert, Dina Kamel, and Amir Moradi. Side-channel attacks from static power: When should we care? In *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 145–150. IEEE, 2015.

[27] Alper Demir and Alberto Sangiovanni-Vincentelli. Time-domain non-monte carlo noise simulation. In *Analysis and Simulation of Noise in Nonlinear Electronic Circuits and Systems*, pages 113–161. Springer, 1998.

[28] Zeynep Toprak Deniz and Yusuf Leblebici. Low-power current mode logic for improved dpa-resistance in embedded systems. In *International Symposium on Circuits and Systems (ISCAS 2005), 23-26 May 2005, Kobe, Japan*, pages 1059–1062, 2005.

[29] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete - or how to evaluate the security of any leaking device. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 401–429, 2015.

[30] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete or how to evaluate the security of any leaking device. *IACR Cryptology ePrint Archive*, 2015:119, 2015.

[31] François Durvaux, François-Xavier Standaert, and Santos Merino Del Pozo. Towards easy leakage certification. In *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th*

International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings, pages 40–60, 2016.

[32] François Durvaux, François-Xavier Standaert, and Nicolas Veyrat-Charvillon. How to certify the leakage of a chip? In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 459–476, 2014.

[33] P. Dutta and P. M. Horn. Low-frequency fluctuations in solids: 1/f noise. *Rev. Mod. Phys.*, 53:497–516, Jul 1981.

[34] Thomas Eisenbarth, Sandeep S. Kumar, Christof Paar, Axel Poschmann, and Leif Uhsadel. A survey of lightweight-cryptography implementations. *IEEE Des. Test Comput.*, 24(6):522–533, 2007.

[35] Julie Ferrigno and M Hlaváč. When aes blinks: introducing optical side channel. *IET Information Security*, 2(3):94–98, 2008.

[36] P Flatresse. Product vision on planar fully depleted technology fd28nm for mobile applications. In *Proceedings of the IEEE International SOI Conference, Napa, CA, USA*, pages 1–4, 2012.

[37] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *International workshop on cryptographic hardware and embedded systems*, pages 251–261. Springer, 2001.

[38] Luca Giancane, Piero Marietti, Mauro Olivieri, Giuseppe Scotti, and Alessandro Trifiletti. A new dynamic differential logic style as a countermeasure to power analysis attacks. In *15th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2008, St. Julien's, Malta, August 31 2008-September 3, 2008*, pages 364–367, 2008.

[39] Hannes Groß, Stefan Mangard, and Thomas Korak. Domain-oriented masking: Compact masked hardware implementations with arbitrary protection order. In Begül Bilgin, Svetla Nikova, and Vincent Rijmen, editors, *Proceedings of the ACM Workshop on Theory of Implementation Security, TIS@CCS 2016 Vienna, Austria, October, 2016*, page 3. ACM, 2016.

[40] Peng Gu, Dylan Stow, Russell Barnes, Eren Kursun, and Yuan Xie. Thermal-aware 3d design for side-channel information leakage.

In *2016 IEEE 34th International Conference on Computer Design (ICCD)*, pages 520–527. IEEE, 2016.

[41] Tim Erhan Güneysu. *Cryptography and cryptanalysis on reconfigurable devices: security implementations for hardware and reprogrammable devices*. PhD thesis, Ruhr University Bochum, 2009.

[42] Wilfried Haensch, Edward J. Nowak, Robert H. Dennard, Paul M. Solomon, Andres Bryant, Omer H. Dokumaci, Arvind Kumar, Xinlin Wang, Jeffrey B. Johnson, and Massimo V. Fischetti. Silicon CMOS devices beyond scaling. *IBM J. Res. Dev.*, 50(4-5):339–362, 2006.

[43] Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In *Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings*, pages 239–252, 2006.

[44] Jip Hogenboom and Lejla Batina. Principal component analysis and side-channel attacks-master thesis. *Principal component analysis and side-channel attacks-master thesis*, pages 536–539, 2010.

[45] FN Hooge. Discussion of recent experiments on 1/f noise. *Physica*, 60(1):130–144, 1972.

[46] FN Hooge, TGM Kleinpenning, and LKJ Vandamme. Experimental studies on 1/f noise. *Reports on progress in Physics*, 44(5):479, 1981.

[47] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[48] Michael Hutter and Jörn-Marc Schmidt. The temperature side channel and heating fault attacks. In *International Conference on Smart Card Research and Advanced Applications*, pages 219–235. Springer, 2013.

[49] E. G. Ioannidis, S. Haendler, A. Bajolet, T. Pahron, N. Planes, F. Arnaud, R. A. Bianchi, M. Haond, D. Golanski, J. Rosa, C. Fenouillet-Beranger, P. Perreau, C. A. Dimitriadis, and G. Ghibaudo. Low frequency noise variability in high-k/metal gate stack 28nm bulk and fd-soi cmos transistors. In *2011 International Electron Devices Meeting*, pages 18.6.1–18.6.4, Dec 2011.

[50] Kengo Iokibe, Tetsuo Amano, Kaoru Okamoto, and Yoshitaka Toyota. Equivalent circuit modeling of cryptographic integrated circuit for information security design. *IEEE transactions on electromagnetic compatibility*, 55(3):581–588, 2013.

[51] Kengo Iokibe, Ryota Higashi, Takahiro Tsuda, Kouji Ichikawa, Katsumi Nakamura, Yoshitaka Toyota, and Ryuji Koga. Modeling of microcontroller with multiple power supply pins for conducted emi simulations. In *2008 Electrical Design of Advanced Packaging and Systems Symposium*, pages 135–138. IEEE, 2008.

[52] Dina Kamel, Guerric de Streel, Santos Merino Del Pozo, Kashif Nawaz, François-Xavier Standaert, Denis Flandre, and David Bol. Towards securing low-power digital circuits with ultra-low-voltage vdd randomizers. In *Security, Privacy, and Applied Cryptography Engineering - 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings*, pages 233–248, 2016.

[53] Dina Kamel, Mathieu Renauld, David Bol, François-Xavier Standaert, and Denis Flandre. Analysis of dynamic differential swing limited logic for low-power secure applications. *Journal of Low Power Electronics and Applications*, 2(1):98–126, 2012.

[54] Dina Kamel, Mathieu Renauld, Denis Flandre, and François-Xavier Standaert. Understanding the limitations and improving the relevance of SPICE simulations in side-channel security evaluations. *J. Cryptographic Engineering*, 4(3):187–195, 2014.

[55] Dina Kamel, François-Xavier Standaert, and Denis Flandre. Scaling trends of the AES s-box low power consumption in 130 and 65 nm CMOS technology nodes. In *International Symposium on Circuits and Systems (ISCAS 2009), 24-17 May 2009, Taipei, Taiwan*, pages 1385–1388, 2009.

[56] Monodeep Kar, Arvind Singh, Anand Rajan, Vivek De, and Saibal Mukhopadhyay. What does ultra low power requirements mean for side-channel secure cryptography? In *34th IEEE International Conference on Computer Design, ICCD 2016, Scottsdale, AZ, USA, October 2-5, 2016*, pages 686–689, 2016.

[57] Momoka Kasuya, Takanori Machida, and Kazuo Sakiyama. New metric for side-channel information leakage: Case study on em radiation from aes hardware. In *2016 URSI Asia-Pacific Radio*

Science Conference (URSI AP-RASC), pages 1288–1291. IEEE, 2016.

[58] Stéphanie Kerckhof, François Durvaux, Cédric Hocquet, David Bol, and François-Xavier Standaert. Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, pages 390–407, 2012.

[59] Nam Sung Kim, Todd Austin, David Baauw, Trevor Mudge, Krisztián Flautner, Jie S Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan. Leakage current: Moore's law meets static power. *computer*, 36(12):68–75, 2003.

[60] Sami Kirolos, Mosin Mondal, Kartik Mohanram, and Yehia Massoud. A model-based technique for efficient evaluation of noise robustness. In *2007 50th Midwest Symposium on Circuits and Systems*, pages 714–717. IEEE, 2007.

[61] Johann Knechtel, Elif Bilge Kavun, Francesco Regazzoni, Annelie Heuser, Anupam Chattopadhyay, Debdeep Mukhopadhyay, Soumyajit Dey, Yunsi Fei, Yaacov Belenky, Itamar Levi, et al. Towards secure composition of integrated circuits and electronic systems: On the role of eda. *arXiv preprint arXiv:2001.09672*, 2020.

[62] Paul C Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Annual International Cryptology Conference*, pages 104–113. Springer, 1996.

[63] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 388–397, 1999.

[64] JS Kolhatkar, LKJ Vandamme, C Salm, and H Wallinga. Separation of random telegraph signals from 1/f noise in mosfets under constant and switched bias conditions. In *European Solid-State Device Research, 2003. ESSDERC'03. 33rd Conference on*, pages 549–552. IEEE, 2003.

[65] Radhouane Laajimi, Ali Ajimi, L. Touil, and A. Bahar. A novel design for xor gate used for quantum-dot cellular automata (qca)

to create a revolution in nanotechnology structure. *International Journal of Advanced Computer Science and Applications*, 8, 2017.

[66] Itamar Levi, Davide Bellizia, David Bol, and François-Xavier Standaert. Ask less, get more: Side-channel signal hiding, revisited. *IEEE Trans. Circuits Syst.*, 67-I(12):4904–4917, 2020.

[67] Itamar Levi, Davide Bellizia, and François-Xavier Standaert. Reducing a masked implementation's effective security order with setup manipulations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 293–317, 2019.

[68] Huiyun Li, A. Theodore Markettos, and Simon W. Moore. Security evaluation against electromagnetic analysis at design time. In *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, pages 280–292, 2005.

[69] François Macé, François-Xavier Standaert, and Jean-Jacques Quisquater. Information theoretic evaluation of side-channel resistant logic styles. In Paillier and Verbauwhede [92], pages 427–442.

[70] Sergej Makovejev, B Kazemi Esfeh, V Barral, N Planes, M Haond, Denis Flandre, J-P Raskin, and Valeriya Kilchytska. Wide frequency band assessment of 28 nm fdsoi technology platform for analogue and rf applications. *Solid-State Electronics*, 108:47–52, 2015.

[71] Sergej Makovejev, N Planes, M Haond, Denis Flandre, J-P Raskin, and Valeriya Kilchytska. Comparison of self-heating and its effect on analogue performance in 28 nm bulk and fdsoi. *Solid-State Electronics*, 115:219–224, 2016.

[72] Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In *Topics in Cryptology - CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings*, pages 222–235, 2004.

[73] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[74] Stefan Mangard, Elisabeth Oswald, and François-Xavier Standaert. One for all - all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.

[75] Stefan Mangard and Kai Schramm. Pinpointing the side-channel leakage of masked AES hardware implementations. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, pages 76–90, 2006.

[76] Ramya Jayaram Masti, Devendra Rai, Aanjhan Ranganathan, Christian Müller, Lothar Thiele, and Srdjan Capkun. Thermal covert channels on multi-core platforms. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 865–880, 2015.

[77] Nele Mentens, Lejla Batina, Bart Preneel, and Ingrid Verbauwhede. A systematic evaluation of compact hardware implementations for the rijndael s-box. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings*, pages 323–333, 2005.

[78] Mentor Graphics Corporation. Eldo User's Manual, Release AMS 2008.2. 2008.

[79] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, pages 278–296, 2004.

[80] Thorben Moos. Static power sca of sub-100 nm cmos asics and the insecurity of masking schemes in low-noise environments. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 202–232, 2019.

[81] Amir Moradi. Side-channel leakage through static power. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 562–579. Springer, 2014.

[82] Amir Moradi, Mohammad Taghi Manzuri Shalmani, and Mahmoud Salmasizadeh. Dual-rail transition logic: A logic style for counteracting power analysis attacks. *Comput. Electr. Eng.*, 35(2):359–369, 2009.

[83] M. H. Na, E. J. Nowak, W. Haensch, and J. Cai. The effective drive current in cmos inverters. In *Digest. International Electron Devices Meeting,*, pages 121–124, Dec 2002.

[84] Katsumi Nakamura. Emc macro-model (leccs-core) for multiple power-supply pin lsi. In *2004 International Symposium on Elec-*

tromagnetic Compatibility, Sendai, Japan (EMC'04/Sendai), June, pages 493–496, 2004.

[85] Maxime Nassar, Shivam Bhasin, Jean-Luc Danger, Guillaume Duc, and Sylvain Guilley. BCDL: A high speed balanced DPL for FPGA with global precharge and no early evaluation. In *Design, Automation and Test in Europe, DATE 2010, Dresden, Germany, March 8-12, 2010*, pages 849–854, 2010.

[86] Kashif Nawaz, Léopold Van Brandt, Itamar Levi, François-Xavier Standaert, and Denis Flandre. A security oriented transient-noise simulation methodology: Evaluation of intrinsic physical noise of cryptographic designs. *Integration*, 68:71–79, 2019.

[87] Kashif Nawaz, Leopold Van Brandt, François-Xavier Standaert, and Denis Flandre. Let's make it noisy: A simulation methodology for adding intrinsic physical noise to cryptographic designs. In *14th Conference on Ph.D. Research in Microelectronics and Electronics, PRIME 2018, Prague, Czech Republic, July 2-5, 2018*, pages 61–64, 2018.

[88] Kashif Nawaz, Dinal Kamel, François-Xavier Standaert, and Denis Flandre. Scaling trends for dual-rail logic styles against side-channel attacks: A case-study. In *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, pages 19–33, 2017.

[89] Kashif Nawaz, Itamar Levi, François-Xavier Standaert, and Denis Flandre. A transient noise analysis of secured dual-rail based logic style. In *2018 New Generation of CAS, NGCAS 2018, Valletta, Malta, November 20-23, 2018*, pages 146–149, 2018.

[90] Svetla Nikova, Christian Rechberger, and Vincent Rijmen. Threshold implementations against side-channel attacks and glitches. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 529–545. Springer, 2006.

[91] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of non-linear functions in the presence of glitches. In *Information Security and Cryptology - ICISC 2008,*

11th International Conference, Seoul, Korea, December 3-5, 2008, Revised Selected Papers, pages 218–234, 2008.

[92] Pascal Paillier and Ingrid Verbauwhede, editors. *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*. Springer, 2007.

[93] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Series in Electrical Engineering. McGraw-Hill, 1991.

[94] M.J.M. Pelgrom, A.C.J. Duinmaijer, and A.P.G. Welbers. Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits*, 24(5):1433–1439, oct 1989.

[95] T. Poiroux, O. Rozeau, S. Martinie, P. Scheer, S. Puget, M. A. Jaud, S. E. Ghouli, J. C. Barbé, A. Juge, and O. Faynot. Utsoi2: A complete physical compact model for utbb and independent double gate mosfets. In *2013 IEEE International Electron Devices Meeting*, pages 12.4.1–12.4.4, Dec 2013.

[96] Thomas Popp, Mario Kirschbaum, Thomas Zefferer, and Stefan Mangard. Evaluation of the masked logic style mdpl on a prototype chip. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 81–94. Springer, 2007.

[97] Thomas Popp and Stefan Mangard. Masked dual-rail pre-charge logic: Dpa-resistance without routing constraints. In *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, pages 172–186, 2005.

[98] Santos Merino Del Pozo and François-Xavier Standaert. Blind source separation from single measurements using singular spectrum analysis. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*, pages 42–59, 2015.

[99] Santos Merino Del Pozo and François-Xavier Standaert. Getting the most out of leakage detection - statistical tools and measurement setups hand in hand. In *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, pages 264–281, 2017.

[100] Jan M. Rabaey. *Digital Integrated Circuits: A Design Perspective.* Prentice-Hall, Inc., USA, 1996.

[101] Francesco Regazzoni, Alessandro Cevrero, François-Xavier Standaert, Stéphane Badel, Theo Kluter, Philip Brisk, Yusuf Leblebici, and Paolo Ienne. A design flow and evaluation framework for dpa-resistant instruction set extensions. In *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, pages 205–219, 2009.

[102] Francesco Regazzoni, Thomas Eisenbarth, Axel Poschmann, Johann Großschädl, Frank Gurkaynak, Marco Macchetti, Zeynep Toprak, Laura Pozzi, Christof Paar, Yusuf Leblebici, et al. Evaluating resistance of mcml technology to power analysis attacks using a simulation-based methodology. In *Transactions on Computational Science IV*, pages 230–243. Springer, 2009.

[103] Francesco Regazzoni, Thomas Eisenbarth, Axel Poschmann, Johann Großschädl, Frank K. Gürkaynak, Marco Macchetti, Zeynep Toprak Deniz, Laura Pozzi, Christof Paar, Yusuf Leblebici, and Paolo Ienne. Evaluating resistance of MCML technology to power analysis attacks using a simulation-based methodology. *Trans. Comput. Sci.*, 4:230–243, 2009.

[104] Mathieu Renauld, Dina Kamel, François-Xavier Standaert, and Denis Flandre. Information theoretic and security analysis of a 65-nanometer DDSLL AES s-box. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 223–239. Springer, 2011.

[105] Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 109–128, 2011.

[106] Akashi Satoh, Sumio Morioka, Kohji Takano, and Seiji Munetoh. A compact rijndael hardware architecture with s-box optimization. In *Advances in Cryptology - ASIACRYPT 2001, 7th International*

Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings, pages 239–254, 2001.

[107] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[108] Danilo Šijačić, Josep Balasch, and Ingrid Verbauwhede. Sweeping for leakage in masked circuit layouts. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 915–920. IEEE, 2020.

[109] Danilo Šijačić, Josep Balasch, Bohan Yang, Santosh Ghosh, and Ingrid Verbauwhede. Towards efficient and automated side-channel evaluations at design time. *Journal of Cryptographic Engineering*, pages 1–15, 2020.

[110] Arvind Singh, Monodeep Kar, Jong Hwan Ko, and Saibal Mukhopadhyay. Exploring power attack protection of resource constrained encryption engines using integrated low-drop-out regulators. In *IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED 2015, Rome, Italy, July 22-24, 2015*, pages 134–139. IEEE, 2015.

[111] Arvind Singh, Monodeep Kar, Sanu Mathew, Anand Rajan, Vivek De, and Saibal Mukhopadhyay. Exploiting on-chip power management for side-channel security. In *2018 Design, Automation & Test in Europe Conference & Exhibition, DATE 2018, Dresden, Germany, March 19-23, 2018*, pages 401–406, 2018.

[112] Lindsay I Smith. A tutorial on principal components analysis. 2002.

[113] François-Xavier Standaert, Tal Malkin, and Moti Yung. Does physical security of cryptographic devices need a formal study? (invited talk). In *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, page 70, 2008.

[114] François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April*

26-30, 2009. Proceedings, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

[115] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 112–129, 2010.

[116] Franco Stellari, Alberto Tosi, Franco Zappa, and Sergio Cova. Cmos circuit testing via time-resolved luminescence measurements and simulations. *IEEE Transactions on Instrumentation and Measurement*, 53(1):163–169, 2004.

[117] Petre Stoica and Randolph L Moses. *Introduction to spectral analysis*, volume 1. Prentice hall Upper Saddle River, 1997.

[118] Petre Stoica, Randolph L Moses, et al. *Spectral analysis of signals*, volume 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

[119] Yuan Taur. Cmos design near the limit of scaling. *IBM Journal of Research and Development*, 46(2.3):213–222, 2002.

[120] C.G. Theodorou, E.G. Ioannidis, S. Haendler, E. Josse, C.A. Dimitriadis, and G. Ghibaudo. Low frequency noise variability in ultra scaled fd-soi n-mosfets: Dependence on gate bias, frequency and temperature. *Solid-State Electronics*, 117(Supplement C):88 – 93, 2016. PLANAR FULLY-DEPLETED SOI TECHNOLOGY.

[121] Kris Tiri, Moonmoon Akmal, and Ingrid Verbauwhede. A dynamic and differential cmos logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proceedings of the 28th European solid-state circuits conference*, pages 403–406. IEEE, 2002.

[122] Kris Tiri and Ingrid Verbauwhede. Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, pages 125–136, 2003.

[123] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation. In *2004*

Design, Automation and Test in Europe Conference and Exposition (DATE 2004), 16-20 February 2004, Paris, France, pages 246–251, 2004.

[124] Kris Tiri and Ingrid Verbauwhede. A digital design flow for secure integrated circuits. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 25(7):1197–1208, 2006.

[125] Ehsan Toreini, Brian Randell, and Feng Hao. An acoustic side channel attack on enigma. *School of Computing Science Technical Report Series*, 2015.

[126] Mark Dolan Torgerson, Timothy John Draelos, Richard Crabtree Schroeppel, Russell D Miller, and William Erik Anderson. Small circuits for cryptography. Technical report, Sandia National Laboratories, 2005.

[127] M. J. Uren, M. J. Kirton, and S. Collins. Anomalous telegraph noise in small-area silicon metal-oxide-semiconductor field-effect transistors. *Physical Review B*, 37(14):8346–8350, may 1988.

[128] Léopold Van Brandt, Denis Flandre, and Jean-Pierre Raskin. Ac compact modelling of low-frequency noise for 28nm cmos technology and beyond.

[129] Léopold Van Brandt, Pascal Bolcato, and Denis Flandre. Generation of Noise Sources for TransientSimulation: a Comprehensive Mathematical Perspective. unpublished, in preparation, 2021.

[130] Aldert Van der Ziel. *Noise in solid state devices and circuits*. Wiley-Interscience, 1986.

[131] Aldert Van der Ziel. Unified presentation of 1/f noise in electron devices: fundamental 1/f noise sources. *Proceedings of the IEEE*, 76(3):233–258, 1988.

[132] L. K. J. Vandamme, Xiaosong Li, and D. Rigaud. 1/f noise in mos devices, mobility or number fluctuations? *IEEE Transactions on Electron Devices*, 41(11):1936–1945, Nov 1994.

[133] Lode K. J. Vandamme and F. N. Hooge. What do we certainly know about $1/f$ noise in MOSTs? *IEEE Transactions on Electron Devices*, 55(11):3070–3085, nov 2008.

[134] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel

attacks: A comprehensive study with cautionary note. In *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, pages 740–757, 2012.

[135] Dennis Wackerly, William Mendenhall, and Richard Scheaffer. *Mathematical statistics with applications*. Nelson Education, 2007.

[136] Piet Wambacq and Willy MC Sansen. *Distortion analysis of analog integrated circuits*, volume 451. Springer Science & Business Media, 2013.

[137] Alexander Wild, Amir Moradi, and Tim Güneysu. Glifred: Glitch-free duplication towards power-equalized circuits on fpgas. *IEEE Trans. Computers*, 67(3):375–387, 2018.

[138] Y. Yano, K. Iokibe, Y. Toyota, and T. Teshima. Signal-to-noise ratio measurements of side-channel traces for establishing low-cost countermeasure design. In *2017 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC)*, pages 93–95, 2017.

[139] Weize Yu, Orhun Aras Uzun, and Selçuk Köse. Leveraging on-chip voltage regulators as a countermeasure against side-channel attacks. In *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015*, pages 115:1–115:6. ACM, 2015.

Acronyms

ABB Adaptive Back-gate Body Biasing. iii, 9, 10

AES Advanced Encryption Standard. 15, 17

BOX Buried Oxide. 8, 9

CMOS Complementary Metal-Oxide Semiconductor. iii, 8, 10–13, 18, 21, 31

DIBL Drain Induced Barrier Lowering. 8

FBB Forward Body/Back-gate Bias. 9

FDSOI Fully Depleted Semiconductor-On-Insulator. iii, 8–10, 13

FPGA Field Programmable Gate Array. 17

LVT Low Voltage Transistors. 8

MOS Metal-Oxide Semiconductor. 13, 14

MOSFET Metal-Oxide Semiconductor Field Effect Transistor. iii, 7–10, 14

NMOS n-channel MOSFET. 12

PDK Process-Design Kit. 8

PMOS p-channel MOSFET. 12

PSD Power Spectral Density. 14, 15

RBB Reverse Body/Back-gate Bias. 9

RTN Random Telegraph Noise. 14

RVT Regular Voltage Transistors. 8

Sbox Sub Bytes. 15, 17

SCA Side-Channel Analysis. 7, 15

SVT Standard Voltage Transistors. 8

XNOR Exclusive-NOR. 18

XOR Exclusive-OR. iii, 15–18