SCMA: Plaintext Classification Assisted Side Channel Spectral Modulation Attacks

Towards Noise-insensitive SCA Attacks...

Moshe Avital[2] and Itamar Levi[1]

[1] Bar-Ilan University, Ramat-Gan, Israel. Email: itamar.levi@biu.ac.il
[2] RAFAEL defense systems inc., Haifa, Israel. Email: mosheavi@gmail.com

Abstract. Side-channel analysis (SCA) attacks manifest a significant challenge to the security of cryptographic devices. It turns out to be generally quite expensive to protect from SCAs (in terms of energy, area, performance etc.). In this work we exhibit a significant change in paradigm for SCA attacks: our proposed attack is quite different from conventional SCA attacks and is able to filter out physical measurement noise, algorithmic noise, as well as thwart various countermeasures, and extract information from the entire leakage waveform as a whole and not only points-of-interest. We demonstrate on measured devices break of masking schemes of orders 2 and 3, supported by a model and also shuffling and dual-rail based countermeasures model; all performed efficiently with the same methodology, and with orders of magnitude less measurements and smaller computation time; underpinning the importance of this form of attack. In essence, in our attack we assume nothing different than a standard side-channel attack, i.e., a known plaintext scenario. However, we further group and classify leakages associated with specific subsets of plaintexts bits. The fact that we group specific (sub-)plaintexts associated leakages, and than in the next stage group or concatenate the associated leakages of these large groups in a predefined ordered sequence (modulation), enables far stronger attacks against SCA protected and unprotected designs. The evaluation-domain or the modulation-domain is the frequency domain in which per frequency it is possible to build a two feature constellation diagrams (amplitude and phase) and construct distinguishers over these diagrams. On top of the methodological contribution of this new SCA, the main observation we push forward is that practically such an attack is devastating for many countermeasures we were used to consider as secure to some level, such as masking or shuffling with large permutation size. As an example, leakage from a third order masked design can be detected with merely 100 leakage traces from the first statistical moment of the leakage as compared to $15 \cdot 10^6$ traces with conventional SCA leakage detection test from the third statistical order.

Keywords: Chosen plaintext · Frequency attacks · Known plaintext · Leakage modulation · Leakage ordering · Masking · SCA · Side-channel attacks · Shuffling · Spectral modulation · Plaintexts grouping

1 Introduction

Side-channel analysis attacks (SCAs) are efficient cryptanalysis methods where adversaries does not only utilize communicated information within a cryptographic protocol and publicly available mathematical properties of the cryptographic algorithm, but also utilizes knowledge related to physical properties of the implementation and so called side-channel leakages, stemming from the implementation of the algorithm.

Most side-channel attacks discussed in literature assume known-plaintext (or ciphertext) adversarial scenarios. Chosen plaintext (or ciphertext) assisted SCA attacks were previously

considered in several scenarios: side-channel collision attacks were proposed and evaluated in [SWP03, SLFP04, Bog07] where collisions in certain internal variables for different executions were detected utilizing side channel information. Some known/chosen-plaintext abilities including precomputations were required. Various attacks which also utilize chosen plaintexts combined with algebraic analysis of block ciphers and *precomputation* of internal values exist (just to name few, [YWQ09, ZWG11]). These attacks utilize destinguishers based on side channel leakages for classification. In essence, *profiled* side channel attacks also work by profiling leakages associated with chosen plaintexts [CRR02, MS16]. These are later used in the attack phase to predict (parts) of the internal secret state of a device. However, no great advantage is given to the adversary by restricting the plaintext space or only building a model for a subset of plaintexts leakages. In fact, generally, the chosen-plaintext profiled scenario is a limiting factor for the adversary as he needs to screen the known plaintexts in the attack phase [SKS09].

In this work in essence we assume nothing different than a standard side-channel attack. I.e., a known plaintext scenario. However, ***we further group and classify leakages associated with specific subsets of plaintexts bits. The fact that we group specific (sub-)plaintexts associated leakages, and later combine the associated leakages of these large groups in a predefined ordered sequence (modulation), enables far stronger attacks against SCA protected and unprotected designs in the modulation-domain which is the frequency domain***. Therefore, such methodologies should be considered while evaluating side-channel security. I.e., the conceptually different approach of leakage modulation assisted by *plaintext grouping*, *sets' ordering* of leakages from within groups and spectral distinguishers, may show surprisingly efficient in breaking some countermeasures and therefore dangerous. The proposed methodology directs the leakage to where it is most vulnerable in the spectrum. We discuss in later sections the implications relating to some countermeasures. For example, it is already interesting to hint here that (e.g.) shuffling based countermeasures [LBS20, VCMKS12, LKF15], which are known to be more sensitive with frequency based alignment and distinguishers [LCS+21, LH20, vWWB11, MvWB11, TGWC18, PHF08], should be much more vulnerable in the proposed attack scenario. Dual rail based countermeasures (or current-flattening techniques) [TV04, LBBS20] would also be much more vulnerable to this form of attack as well owing to the cumulative signal properties of the attack moving the problem to the spectrum domain with a (far) less noisy signal. Owing to the spectral modulation and mixing of different leakages sets (grouping and ordering), which is a non-linear transformation, information which appear naturally in higher statistical moments of the leakage appear directly in the transform space (amplitude and phase). As shown below it makes masking based countermeasures [CJRR99, ISW03, CGLS20] sensitive to leak in the first statistical order with only few traces, lower than the theoretical security order (d). We demonstrate these properties and how attack data complexity reduces exponentially with d, by using our modulation and distinguisher. E.g., whereas a standard T-test require about 18 million traces with a given noise-level and $d = 3$ to get significant information from the 3^{rd} moment, our distinguisher and modulation detects leakage with about 1000 traces from the first moment leakage. Another feature which makes this possible is the fact that noise influence becomes negligible (and noise is very important with masking). This is because: (1) when taking a very long trace (a concatenated set of traces) and performing an FFT over it, the spectral characteristics extracted in each freq. performs an averaging effect/eliminating noise (2) our proposed data-augmentation technique by permutations within the leakages sets (within class) can be used to generate surprisingly large number of leakages from tiny leakages sets; this later property can not be achieved with standard SCA attacks and is indeed very important in our proposed methodology.

Notably, the scenario of *adaptive* selection of the plaintexts in an SCA context was evaluated in [VCS10], supported by theoretical model and experimentation. We note that

the same gains achieved by the adaptive adversary equally apply in our context.

Our Contribution: We first provide a set of different and powerful techniques for SCAs and security-evaluation. These are all novel leading to a game-changing SCA scenario. As a more holistic discussion, we show another significant ability which exist in the adversary's tool-set. Namely, *the ability to shape the characteristics of the physical leakage*. This is achieved by jointly utilizing these three proposed techniques: (i) *subset-grouping and classification* of small number of plaintext (or ciphertext) bits, (ii) efficient leakages sets *ordering* or modulation in accordance with (i), and then (iii) applying dedicated *spectrum based distinguishers*. We demonstrate that by using such a methodology and crafted set of techniques, several very dangerous abilities are given to the adversary $\mathcal{A}$:

1. **Leakage modulation:** sensitive information can be tuned to leak in any slice of the spectral domain the adversary wishes. This is possible due to utilization of leakage classification, grouping and ordering techniques in accordance with plaintexts bits. Ideally, such tuning can target a noiseless/activity-less spectral slice. I.e., directing the leakage to where it is most vulnerable.
2. **Cumulative signal:** extracting cumulative information on internal variables from the entire trace *waveform*, not only a subset of time-samples or points-of-interest as traditionally performed with conventional SCA attacks and so-called multivariate attacks or combining functions. The latter suffer from complexity challenges (e.g.,) as the number of variables or points-of-interests increases. Our proposed methodology is straight forward, low-complexity and integrates cumulative information or signal from the entire trace on all correlated internal-variables.
3. **Noise independence:** in this work we examine a distinguisher which modulates the side channel leakages. However, before applying our distinguisher an important step in our attack consist with grouping of leakages in accordance with known subsets of plaintexts bits (as denoted by subset grouping). The specific modulation into the spectrum domain is shown to be very efficient to filter the noise in the leakage to the verge it is quite hard to be utilized as an efficient countermeasure. In fact, we demonstrate how our proposed data-augmentation technique by leakages-sets permutations can provide means to remove noise in unprotected designs scenarios, and in protected designs scenarios such as shuffling, hiding and masking.
4. **Countermeasures included:** we demonstrate the aforementioned properties on actual measurements for unprotected software Tiny-AES traces, and we show that protected designs are becoming an easy targets for such an adversary, $\mathcal{A}$: namely we model shuffling and hiding (by dual-rail) leakages showing their in-ability to conceal secrets with our modulation and distinguisher, and we model high-order Boolean masking leakages up to the fourth order *showing how our modulation and distinguisher lower the effective security order from the theoretical d to leakage in the first statistical moment* with very few measurements. The latter highlight comes with a contribution of relevant and adapted T-test, relevant to our modulation and the spectrum samples. We also demonstrate it on actual leakages from $d = 2$ and 3 masked HW scenario over an FPGA device of the Present algorithm as an example.

Paper organization. The manuscript begins with a technical background in Section 2 where we briefly elaborate on standard and existing SCAs, and needed evaluation metrics. In Section 3 we layout the proposed methodology and specifically discuss the modulation procedures in Sub-section 3.1, and formulate our tailored spectral distinguishers characteristics in Sub-section 3.2. In Sub-section 3.3 we elaborate on the powerful data-augmentation technique by permutations. In Section 6 we follows with modeling and simulating our attack on an unprotected model, shuffling and dual-rail leakage models and protected masking leakage model, including mathematical model of the modulated leakages. Finally, in Section 5 we show several concrete examples over software and hardware measured date.

2 Background

Strength and Limitation of Current Time-Domain SCAs: Statistical side channel attacks (Differential or Correlation power or electromagnetic side channel attacks), or as termed in the jargon DPAs, are traditionally uni-variate and extract information from a single to several proximate time samples in the leakage and from a single internal variable computed within the algorithm [BCO04, KJJ99, DPRS11], targeting low(er) adversarial computational effort with relatively large data-complexity (# traces). However, clearly a univariate methodology can extract little information from the huge traces collected by the adversary. On the contrary, several reports suggested to perform a multi-variate attack [BCPZ16, OM93, BGH+17, Riv08, LPR+14]. Such approaches incur (far-)larger computational effort and to be successful require some adversarial knowledge; however, indeed reduce data-complexity. In between, several dimensionality reduction techniques [CDP16, BGH+15] were proposed which aim to balance this trade-off, clearly at the cost of quality. Especially in the masking context where several splits or shares of the same internal variables are processed in (e.g.,) different time samples, for example utilizing combining-functions [FMPR11, CPRR14], utilizing transforms [DSEA+12], filtering techniques which group/combine information from different time samples [SWL21, BFP22, TGWC18] etc.

Though relatively efficient, none of these time-domain based attacks have the potential to extract all pieces of information related to or correlated with some hypothesized internal-variable from the entire trace waveform jointly in a rather agnostic way: i.e., requiring little knowledge on the implementation and where in time to search for pieces of information (points of interest, POIs). In this work we propose the first step in achieving this property with low computational effort. We demonstrate attacks which are far stronger below.

Limitations of Current Frequency-Domain SCAs: In the literature, differential or correlation frequency analysis attacks were proposed (D/CFA) [MG10, Tiu05], where leakages are transformed to the frequency domain by fast-furrier transform (FFT) or wavelet based transform, to the wavelets domain [DSEA+12].

The main limitations of such approaches lies in the following facts

1. They operate on each (short) leakage trace independently, where the sampling theory (Nyquist) tells us that the representation of (sampled) continuous signals with harmonics is limited and will encompass errors for (rather) short traces anyway.
2. Transforms are not accurate in traces boundaries (i.e., initial/final rounds)

The proposed approach in this paper is inherently different: the proposed transform operates on (public) plaintext[1]-dependent concatenated sets of traces and then uses tailored modulations, i.e., interleaving leakages in an anti-symmetric duty-cycled fashion among data-dependent classes. This modulation enhance the ability of the transform to extract information accurately and cumulatively from many computations jointly modulated, and not just one from one leakage trace. In the approach proposed here, we first make sure that sampling theory (Nyquist) hold for the entire spectrum of information we are interested in, and by so we capture information from the entire leakage waveform which is modulated to a low-noise spectral region and chosen harmonies. The information is captured in the encryption frequency, f_{enc}, which is shifted (modulated) with duty-cycled (D.C) interleaving of classes associated leakages, and the number of rounds the adversary wants to consider #R from each trace.

2.1 Necessary Side Channel Attacks (SCAs) Background and Notations

Side channel attacks have repeatedly demonstrated the sensitivity of implemented cryptographic schemes. As such, current National Institute of Standardization and Technology

[1] we assume a forward attack direction from the plaintext, as typically done in literature, though the backward direction from the ciphertext is equally applicable

(NIST) competitions for future symmetric-key, e.g., Authenticated-Encryption [TMC+21] and public-key Post-Quantum schemes [AASA+20], are considering SCA security as important aspects. In this subsection we briefly recall some of the necessary SCA basics and details needed for this manuscript on SCA security evaluation-metrics. In more details, side channel attacks enable the extraction of secret values manipulated by the hardware by exploiting the dependencies of secret-key dependent computations and some physically measurable quantity. Most reports focus on side-channel leakage measured through power or electromagnetic radiation channels, owing to their ease of access and rather high signal to noise ratio. These leakages originate mainly as physical outcomes of dynamic (switching) current dissipation of microelectronic devices. Correlation or Differential power analysis (C/DPA) [KJJ99, BCO04] are powerful side-channel attacks that follow a divide-and-conquer approach: an estimation on distinct parts of the key (denoted by sub-keys) takes place, called hypothesis, and these hypotheses are checked for correlations with the measured leakage from the device through multiple tests.

In simulated environments we typically estimate a leakage model for some intermediate values manipulated by the device. A leakage model is aimed at representing to some extent the actual physical behaviour measured in the SCA paradigm; i.e., the leakage owing to internal values manipulation. For a commonly practiced leakage model, which typically nicely represents software implementations leakage, we can consider the Hamming Weight (HW) leakage model by which an intermediate variable y of n-bits leaks: $\alpha \cdot \mathrm{HW}(y) + \beta + \mathcal{N}(\mu, \sigma^2)$. Where, the α and β factors may represent some signal scaling and DC offsets. $\mathcal{N}(\mu, \sigma^2)$ is the modeled noise distribution owing to internal factors within the device and external parameters such as environmental influences. Within simple first-order estimation models, we typically assume a Gaussian additive and independent noise samples.

Similarly to traditional cryptographic models, we assume one end of the communication is exposed to an adversary: decryption-leakages measurements are associated with some known ciphertexts (or encryption-leakages with their associated plaintexts). Along the last few decades, the most powerful SCA attacks were statistical, meaning many such (plaintext and encryption-leakage) pairs were retrieved by the adversary, $\mathcal{A}$. Then, $\mathcal{A}$ follows a modeling phase of a key-dependent internal computation in the algorithm, and further models the effect this value has on the leakage. Then, a statistical distinguisher is used, with the goal of classifying the while eliminating the large measurements noise.

Model-Based Attacks: Let's assume that a device performs encryption and that any randomness used by the protocol is public. Therefore, with conventional CPA [BCO04], we assume multiple measurements are available to the adversary under the same secret-key, e.g., considering a symmetric-key block-cipher instantiated. Let $l_{x,k}$ be a leakage trace measurement under a plaintext/key $x = \{x_0||x_1||...x_{15}\}/\ k = \{k_0||k_1||...k_{15}\}$ of n-bits where for example each x_i or k_i represents one byte of say n =128 bits. To perform a CPA attack, one should choose a target computation to hypothesize (in the case of an AES, typically the first round Sbox output). That is, some logical manipulation of the known plaintext (byte) by a deterministic function and the secret-key (byte), e.g., the target intermediate value $y_i = \mathsf{Sbox}(x_i \oplus k_i)$. Once a specific intermediate value is chosen for an attack, the adversary builds/computes an hypothesis table per each hypothesized secret-key $k_i \in \{0,1\}^8$ and all x_i related possible measurements of size N_{tr}. The leakage model for example can follow directly the HW of the intermediate hypothesized y_i value as discussed above. Next, a distinguisher can be used to find the hypothesized model which depends on the secret-key which most correlates with the measured leakage, e.g., a simple and commonly used distinguisher is the Pearson Correlation Coefficient, $\rho_{k^h} = \rho(\mathbf{y}_i^{k^h}, \mathbf{l}_{x,k})$, where $\mathbf{y}_i^{k^h}$ is $\mathbf{y}_i$ computed under an hypothetical key k^h, i.e., $\mathbf{y}_i^{k^h}$ =$\mathsf{Sbox}(x_i \oplus k^h)$. Eventually, the secret-key (byte) k^*, which maximizes the correlation, is chosen, namely $k^* = \mathrm{argmax}_{k^h}\ \rho_{k^h}$. In practice, each leakage trace/measurement is a vector over time $l^t_{x_i,k_i}$ where $t \in \{0, \ldots, \#\mathrm{Samples}\}$ and the correlation in the equation takes the maximum value found over time per hypothesized

key, estimated in a point-of-interest (POI) in time where the secret value y_i is being manipulated by the device [SMY09, LPB+15].

Side Channel Signal to Noise Ratio, SNR: The SNR metric is important for characterization of the leakage signal and the noise level in the measurements for evaluation of the ability to extract sensitive information. The SNR is a very low computational effort metric which was first proposed by Mangard et-al. [Man04], namely the SNR applied to the SCA context. When applied on the measured leakage it can aid in finding a Point of Interest (POI), a point in time that may be leaking information in correlation with some intermediate computed and sensitive variable, as well as indicate the strength of the leakage (information sense). The SNR is calculated for each time sample and requires multiple leakage samples for each (say) known plaintext byte x_i and key bytes k to obtain the variance and the expected values of the leakage per such class: $\mathrm{SNR} = \frac{\mathrm{Var}_{x_i,k}(\mathrm{E}[l^t_{x_i,k}])}{\mathrm{E}_{x_i,k}(\mathrm{Var}_i[l^t_{x_i,k}])}$. SNR was used here mainly owing to popularity in literature and as a baseline comparison.

Leakage detection by T-test: In this paper, we test for the existence of leakages in high statistical moments up to the d^{th} moment, denoted as $\hat{M}^d_s$. Moments are computed on a subset of the leakage samples. The samples are grouped by an internally processed secret value s (either '0' or a '1'); i.e., over l^t_s, the leakage time sample ($t \in \{0, ..., \#\text{samples}\}$), corresponding to different outcome manipulation of s. For the 2^{nd}-order, the second-order central-moment, $CM^{2,t}_s = E((l^t_s - \mu)^2)$, is used instead of the raw moment, M; and for higher orders ($d > 2$), the standardized moment is used, $SM^{d,t}_s = E((\frac{l^t_s - \mu}{\sigma})^d)$. Where, μ and σ are the populations' means and standard deviations, respectively; μ and σ operate on the entire vector of observations in a set per time sample $\mathbf{l}^t_s$.

Our analysis is based on the Test Vector Leakage Assessment procedure from Cryptography Research, CRI [CMG+, GGJR+11] utilizing Welch's (two-tailed) T-test [Wel47]. It is computed on two input sequences (Set_0 and Set_1). In this work, we compare two classes of leakages with so-called specific "*fixed* vs. *random*" [Sta19, DS16] tests to detect leakages, using the following T-test statistic[2]:

$$T_{val} = \big(\mathrm{E}(SM^i_{Set_0}) - \mathrm{E}(SM^i_{Set_1})\big)/(\mathrm{Var}(SM^i_{Set_0})/|Set_0| + \mathrm{Var}(SM^i_{Set_1})/|Set_1|)^{0.5}, \quad (1)$$

3 The Proposed Methodology

The proposed approach is generally based on a different analysis of side-channel information, compared to conventional attacks. In conventional side channel attacks an attacker aims to reveal leaked information by trying to classify a specific time range of the sampled traces (i.e., POIs) using some model on the desired intermediate variable, as shown in Figure 1a. In contrast, the proposed approach considers the entire sampled traces. The approach relies on manipulating the sampled sets of traces (the recorded data for later analysis attack phase), and moving the security evaluation phase to the frequency domain, extracting much more enhanced information. The methodology can be divided into two main parts: (i) the recording part, (ii) the modulation part. The recording part includes sampling and gathering sufficient amount of traces related to known plaintext. The modulation part include several steps such as classification, grouping, interleaving and F-transforming. The detailed methodology is described in next sections.

3.1 Modulation

The modulation part includes the signal processing procedure implemented on the recorded traces for known plaintext (e.g., power consumption, EM measurements), aiming eventually to characterize the information leakage of the analyzed device. The conventional data analyses on the time-domain sampled traces such as CPA, t-test etc. assume (ideally)

[2] Computed by the generalized fast implementation from [SM15]

a single point in time that leaks information. Usually some post-processing on these traces are necessary such as alignment, to achieve accurate synchronization between the traces. In contrast, the modulation process of the proposed approach consists of quite different data analysis, taking into account much more information distributed continuously throughout many clock cycles of each trace. In the following, we describe the main operations that make up the overall modulation part. Additionally, this procedure is illustrated in Algorithm 1.

Classification: The first operation of the modulation is the classification $\boldsymbol{\mathcal{C}}$. The purpose of the operator $\boldsymbol{\mathcal{C}}$ is to rearrange the indices of the sampled set of traces, corresponding to the internal variable y_i^*. The operator receives N such internal y_i^* vectors (for example created by random N *plaintexts*), the number of the observed\classification bits, b, out of the entire 128-bit, and the location, ℓ, of the observed bits, relative to the MSB. The operator in turn outputs classified subsets of the internal variables, $\boldsymbol{\{CS\}}$, where each subset $\boldsymbol{CS}$ contains the internal variables that have the same value of the observed bits b, at the same location ℓ, as described in Equation 2.

$$\boldsymbol{\mathcal{C}}\Big(\{y_i^*\}_1^N, b, \ell\Big) = \{\boldsymbol{CS}_1, ..., \boldsymbol{CS}_{2^b}\} \tag{2}$$

As a consequence, the operator $\boldsymbol{\mathcal{C}}$ outputs 2^b classified subsets, where each can contain a different number of data, defined as $N_1, N_2, ..., N_{2^b}$, such that $N = \sum_{i=1}^{2^b} N_i$. The left side of Figure 1b shows an illustration of the classification operator for N internal variables, $b = 3$, and $\ell = 0$. In this example the 3 observed bits are the 3 MSB bits of the internal variables. The operator $\boldsymbol{\mathcal{C}}$ outputs 8 classes, each corresponds with 3-bit values $\{0, 1\}^3$.

Grouping-Interleaving: The second step of the modulation part is the grouping-interleaving operator, $\boldsymbol{\mathcal{GI}}$, that comes after the classification operator. This operator consists of grouping and interleaving operations. The first portion starts with setting a predetermined number of groups, g, each is set to a predetermined size, s. These degrees of freedom parameters allow us to control the statistical significance between different classifications, and as a result, to control the amount of the leaked signal compared to the noise. Predetermined large g, s parameters might increase the leaked signal, on the expense of computing time and available memory. The successive portion of the grouping is the interleaving. This part interleaves the measured traces (e.g., power, EM) to each of the g groups, according to the classification. The classes created during the classification part are divided to pairs of classes, in an anti-symmetric manner. Meaning, a class relating the observed bits $b_1 b_2 ... b_b$ is paired with the class relating the anti-symmetric observed bits of $\bar{b_1}\bar{b_2}...\bar{b_b}$. In such a way, there are 2^{b-1} pairs of classes. For each, the interleaving operator assigns the measured traces according to the indices of the internal variables from both classes of the pair, using a predefined periodicity, so-called $duty - cycle(d.c)$. For example, the interleaving operator with $d.c = 2$ assigns one trace corresponding to an index of an internal variable to a group from the first class of a pair, and then one trace corresponding to an index of an internal variable from the other class of the pair, and so on. The interleaving operator with $d.c = 3$ assigns one trace corresponding to an index of an internal variable to a group from the first class of a pair, and then two different traces corresponding to two different indices of an internal variable from the other class of the pair, and so on. This part is repeated for 2^b times, where b is the number of the observed bits, for all the g groups. As a result, the operator outputs anti-symmetric interleaved groups of the index of an internal variable, $\boldsymbol{\{ASG_i\}_1^s}$, where $i = 1, ..., g$, as described in Equation 3.

$$\boldsymbol{\mathcal{GI}}\Big(b, g, \{\boldsymbol{CS}_1, ..., \boldsymbol{CS}_{2^b}\}, s, d.c\Big) = \Big\{\{\boldsymbol{ASG}_1\}_1^s, \{\boldsymbol{ASG}_2\}_1^s, ..., \{\boldsymbol{ASG}_g\}_1^s\Big\}_{pair_1}^{pair_{2^{b-1}}} \tag{3}$$

The middle of Figure 1b shows an illustration of the grouping-interleaving operator for $b = 3$ and $d.c = 2$. Note that the group size s can be set to a very large number, as it acts like a permutation of two classes arrangement. For instance, assuming we have

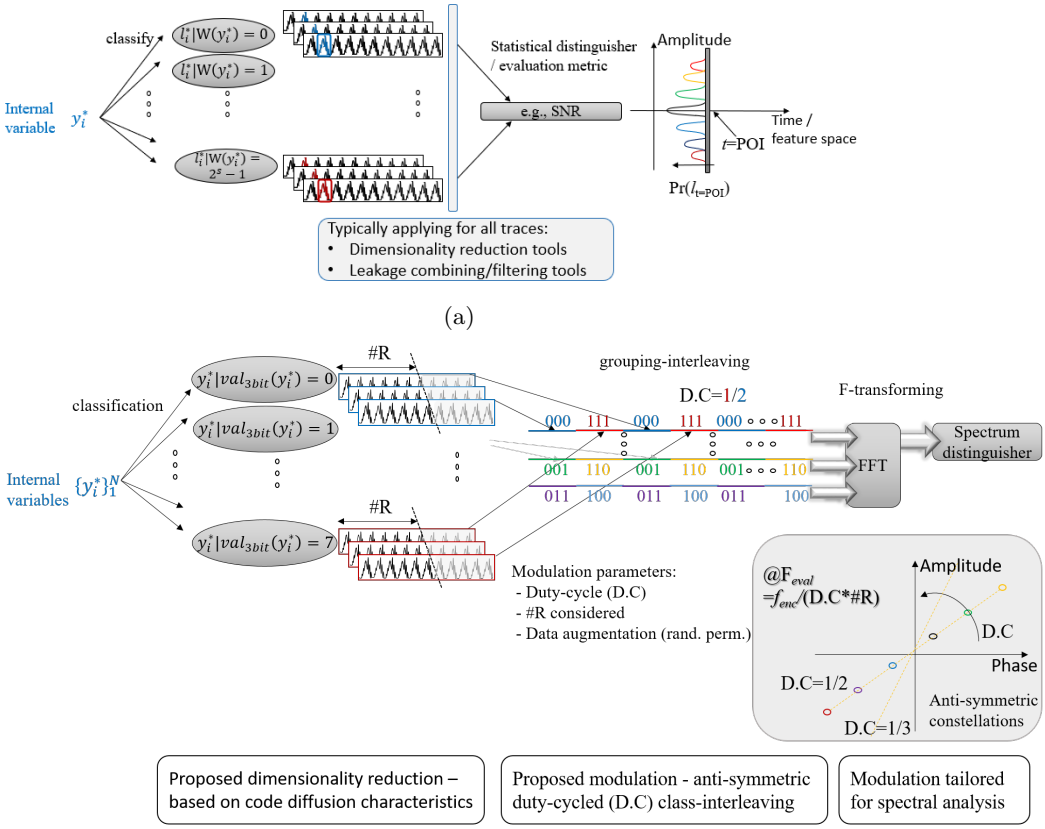


Figure 1: Side-Channel attacks apparatus: (a) Conventional Side Channel attack. (b) Side Channel Modulation Attack apparatus (SCMA).

$N = 1000$ random plaintext, and therefore 1000 internal variables y_i^*, and assuming that for $b = 3$ the first classified subset CS_1 and its anti-symmetric classified subset CS_8 are of size 100. Then the the set size s of each of the groups can be predefined to any value we deem, as there are $(100!)^2$ interleave options.

Code diffusion properties of a ciphers vary and they mainly affect parameters like e.g., the branch number of diffusion/linear layers. Generally, in the case of our discussed attack, leakage is extracted jointly from the entire trace (as elaborated below). Therefore, an adversary has the ability to balance algorithmic/diffusion noise by excluding leakage in deep rounds by the parameter, #R as illustrated in middle Figure 1b, as discussed below this exclusion generates a tractable influence of the modulated leakages featuring a classifiable pattern in the constellation-diagram in a different frequency of interest, as formulated and discussed below, and illustrated in the bottom right part of Figure 1b.

F-transforming: The third step of the modulation part is the F-transforming operator, $\boldsymbol{\mathcal{F}}$, which transforms the interleaved traces prepared in previous part to the frequency domain. At first, Each of the g anti-symmetric groups $\boldsymbol{ASG}_i, i = 1, .., g$, created by a specific classification subset, is concatenated to a single vector. For each pair of anti-symmetric classes, the operator $\boldsymbol{\mathcal{F}}$ transforms (e.g., by using the FFT algorithm) these g concatenated time-domain vectors to g frequency-domain data, consisting *Amplitude* $\boldsymbol{A(f)}$ and *Phase*

Algorithm 1 Leakage Modulation procedure

1: **Recording:** Store a series of N time-domain leakage traces corresponding to N random *plaintext*. **Configuration:** determine the modulation params.: $g, b, d.c, s, \ell$.
2: **Modulation- classification:** rearrange indices of the N leakages according to b, ℓ.
3: **Modulation- loop:** for each pair of anti-symmetric 2^b classes do:
4: **Grouping-Interleaving:** create g groups of size s, and interleave the measured traces in the groups in an anti-symmetric manner according to $\ell, d.c$ parameters and the internal y_i^* variables.
5: **F-transforming:** concatenate each of the g groups and transform them to g frequency-domain *Amplitude* and *Phase* $(A, P)(f)$.
6: **Generate *constellation diagrams*.**
7: **Extract information @chosen** f by operating a distinguisher over the diagrams.

$\boldsymbol{P(f)}$, as described in Equation 4.

$$\boldsymbol{\mathcal{F}}\left(\left\{\{\boldsymbol{ASG}_1\}_1^s, \{\boldsymbol{ASG}_2\}_1^s, ..., \{\boldsymbol{ASG}_g\}_1^s\right\}_{pair_1}^{pair_{2^b-1}}\right) = \left\{\boldsymbol{(A,P)}(f)_1, \boldsymbol{(A,P)}(f)_2, ..., \boldsymbol{(A,P)}(f)_g\right\}_{pair_1}^{pair_{2^b-1}} \quad (4)$$

Building *constellation diagrams* for each of the g obtained spectrum $\boldsymbol{(A,P)}(f)$ for each of the 2^b classes, can be a powerful distinguisher of the leaked information. Next we show that the modulated classes are arranged in the constellation diagram approximately along a straight line, where the different amplitudes correspond to different classes, as illustrated in the lower-right part of Figure 1b. This line is tilted with some angle which is impacted directly by the $d.c$ parameter. Spectra of some class that contain different groups will be located approximately on the same spot of the constellation with some differentiation due to different permutations between the groups (i.e., algorithmic noise).

It is important to emphasize the significant advantage of the proposed approach, which relates to information accumulation from the **entire** time points of the sampled trace, as illustrated in Figure 2. In general, conventional analysis performed within attacks considers *specific* leakage values correlating/matching at independent POIs. As a result, even if considering the entire time points of the traces, these attacks do not take into account the information that may leak along the time (i.e., dependency of different time points) due to the *diffusion* characteristic of (e.g.,) an SPN network or a Sponge. In contrast, the modulation process of the proposed technique accumulates information from the entire trace, and thus considers diffused (and even small) information that may exist in a deep level of the SPN. In addition, this technique has the ability to utilize the degrees of freedom, such as $d.c, b, g, s$, which allow us to increase the signal of the leaked information and significantly reduce the noise. Moreover, the specific anti-symmetric class-interleaving, as illustrated in Figure 2, generates a scenario where (e.g.,) the cumulative affect of some b bits of the leakage highlighted by blue waveform (and blue code-diffusion shape on the SPN structure) is interleaved with an opposite polarity leakage in red highlight (and red code-diffusion shape on the SPN structure). This anti-symmetric interleaving amplifies significantly the separability and classification abilities of the leakage. The FFT operation jointly modulates information from all internal values (regardless of their specific values), and from all leakage time points. The proposed distinguishers are tailored to this extractor.

3.2 Spectral Distinguisher and Leakage Shaping

As mentioned, the information leakage is modulated and can be observed in the frequency domain, utilizing the *Amplitude* and the *Phase*, $\boldsymbol{(A,P)}(f)$, of each class and

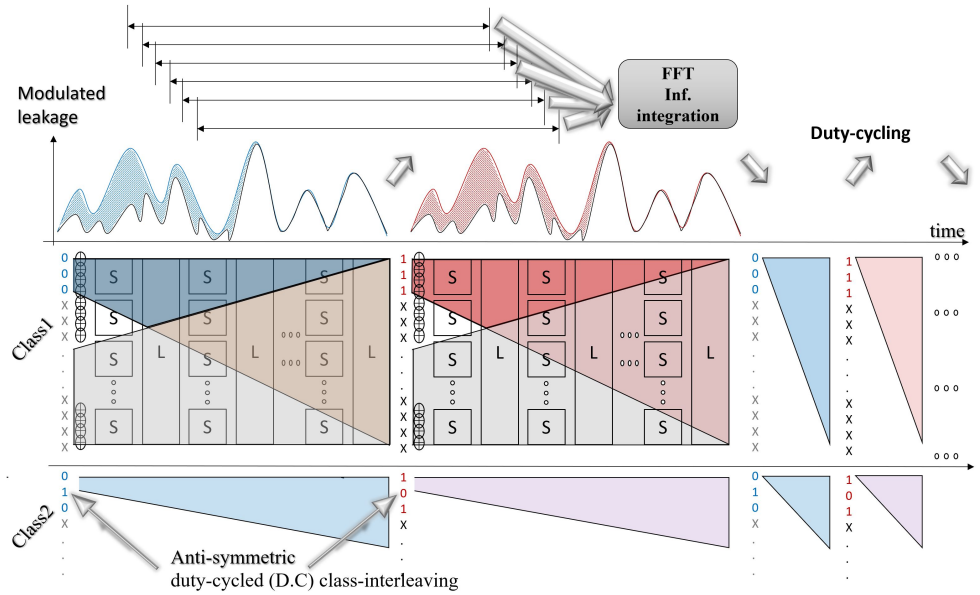


Figure 2: Leakage modulation and information accumulation.

each group. The modulation process results in several frequencies that carry the information. These frequencies include the encryption frequency and its multiplications. The number of frequencies that carry information, $|f_{info}|$, depends on the duty cycle (d.c) and the trace's number of samples n, and can be calculated using Equation 5.

$$|f_{info}| = \left\lfloor \lfloor \frac{n \cdot d.c}{2} \rfloor \cdot (1 - \frac{1}{d.c}) \right\rfloor \tag{5}$$

For each such frequency f_{info}, the corresponding *constellation diagram* containing the $(\boldsymbol{A}, \boldsymbol{P})(f_{info})$ of all 2^b classes can constitute as a spectral distinguisher. As a consequence of the modulation operator, the 2^b points are arranged approximately along the diagram diameter in which its rotation angle, ϕ_{rot} depends on the examined f_{info} and the $d.c$. Equation 6 describes the rotation angle in degrees.

$$\phi_{rot} \equiv 360 \cdot \left(1 - \frac{1}{d.c}\right) \cdot f_{info} \pmod{360} \tag{6}$$

The extent of the ability to separate between each set of g points on the diagram (related to the g groups), determines the strength of the distinguisher. Figure 3 presents an example of four *constellation diagrams* relating to the same informative frequency[3]. The points in each diagram represent the $g = 16$ groups for each one of the $2^{b=3}$ classes. The sub figures of Figure 3 differ in the duty cycle, such that $d.c = 2, 3, 4, 5$ relate to Figure 3(a,b,c,d), respectively. As can be seen by this distinguisher, the eight different classes can be separated in all these cases (circled in dashed lines), and a higher significance is obtained as the $d.c$ is smaller. However, an improved distinguisher for higher $d.c$ values can be achieved by increasing the set size s of each group. Furthermore, this modulation process can be treated as a *leakage shaping* mechanism. By using this algorithm, an attacker can build an efficient strategy and actually navigate the attack to a convenient environment, e.g., a much less noisy frequency area. Meaning, the attacker can control the specific frequencies that carry the leakage information, by defining the desired $b, n, d.c$ parameters, and then, depending on the available resources, s\he can control the grouping g and the set size of each group s in order to improve the separability of the distinguisher.

[3]In the next section we relate to the modeling details of the leakage related to the figure which is only given here as an example

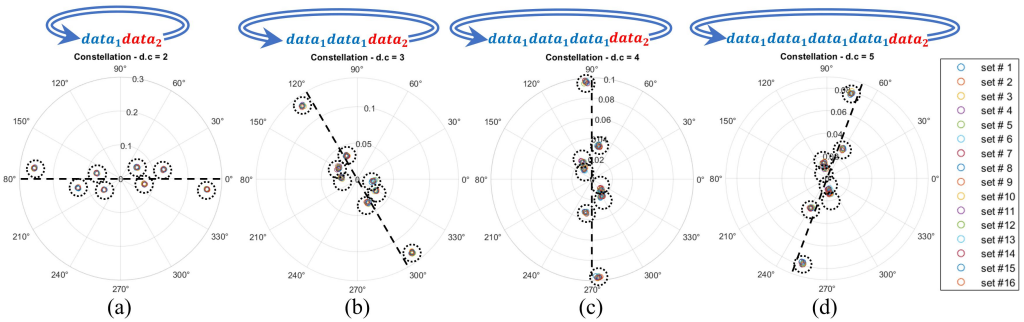


Figure 3: Leakage modulation and constellations: (a) d.c=2 (b) d.c=3 (c) d.c=4 (d) d.c=5

The informative *constellation diagram* allows us to efficiently define distinguishers. As previously explained, since the modulation process enables an attacker to control the signal and the noise of the information leakage, an appropriate SNR for the modulation process is defined, namely, SNR_{mod}. The SNR_{mod} definition is described in Equation 7.

$$SNR(f)_{mod} \triangleq \frac{\mathrm{Var}\Big(\mathrm{E}\big[A(f)\cdot\big(\cos(P(f)-\phi_{rot})+\sin(P(f)-\phi_{rot})\big)\big]\Big)}{\mathrm{E}\Big(\mathrm{Var}\big[A(f)\cdot\big(\cos(P(f)-\phi_{rot})+\sin(P(f)-\phi_{rot})\big)\big]\Big)} \tag{7}$$

Similar to the SNR in context of cryptography, this equation takes into account the distances between different classes as well as distances between the same classes of different groups. Note that A and P are the *Amplitude* and the *Phase* calculated using Equation 4, and ϕ_{rot} is the rotation angle of the *constellation diagram* calculated in Equation 6.

3.3 Powerful Data Augmentation Technique by Permutations

As described in the previous section, the information in the spectral domain can be shaped and allocated to a convenient frequency slice by an attacker setting predefined parameters. A distinguisher can operate on this spectral representation (i.e., constellation) generated by the $\mathcal{F}$ transform (such as FFT algorithm) of the synthetic concatenated traces of complimentary anti-symmetric pairs of classes. Therefore, the distance between the points in the spectral constellation that belong to the same pair of classes but to different groups, depends on the lengths of the concatenated traces. The longer they are, the smaller distances between different constellation points associated with different sets of classes pairs. The length of the concatenated traces depends on the set-size parameter s, as well as the trace's number of samples n and the duty-cycle $d.c$. Low values of s lead to noisy signal or noisy spectral constellation which hardens the attack. On the other hand, high values of s strengthen the signal and weaken the noise, and hence increase the SNR_{mod}. Therefore, it allows an attacker to build an efficient distinguiser that separates different classes much more easily. Moreover, very high values of s, i.e., *Data Augmentation*, can be achieved using quite small set of traces, by performing *permutations*. In other words, data augmentation can be achieved by a huge number of available permutations within a group, each corresponds to a specific concatenation between a pair of complementary anti-symmetric classes. By this data augmentation, an attacker can significantly increase the efficiency to extract information. Figure 4 shows an example that demonstrates the *Data Augmentation* by increasing the set size s of each group. The figure presents four constellations relating to a frequency that carries information with $d.c = 2$, and to three classification bits, i.e., $b = 3$ that yields 8 classes. Different set sizes $s = 2^{13}, 2^{16}, 2^{19}, 2^{22}$ relate to Figure 4(a,b,c,d), respectively. As can be noticed, higher set size s increases the ability of an attacker to distinguish between different classes.

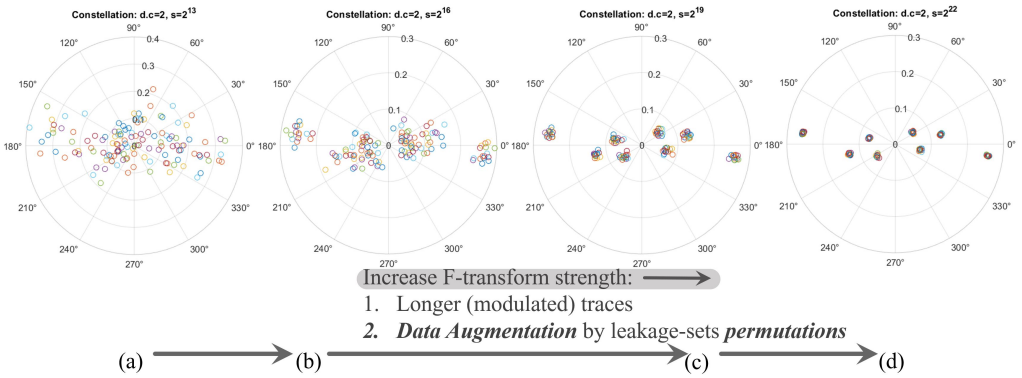


Figure 4: Permutation-based data augmentation for different set size s. (a) $s = 2^{13}$. (b) $s = 2^{16}$. (c) $s = 2^{19}$. (d) $s = 2^{22}$.

Computational complexity: Taking the SNR metric example, a conventional SNR computation is performed over (say) N measured traces, where a classification is made on b bits of the desired intermediate variable. On the other hand, the modulation-based SNR_{mod} analysis requires $d.c$ groups of size s for the interleaving operation, for each of the g groups. For achieving high enough SNR_{mod} values (generally, as mush the adversary wishes), a large set-size s is required. That is, in order to obtain a required SNR_{mod} value, the number of the measured traces N has to be significantly increased, resulting in increased computational complexity. *However, as described previously, increasing the set-size s is efficiently performed by using permutations on a small number of measured traces N, turning this discussion meaningless.* Assuming N measured traces, and b attacked bits, the number of the permutations in each class is approximately $(\frac{N}{2^b})!$. Therefore, as long as $(\frac{N}{2^b})! \gg s$ (which can be easily achieved for example by setting $b = 3$ and $N = 100$), an attacker can increase significantly the set-size s and obtain a desirable SNR_{mod}, highlighting the augmentation and noise-insensitivity properties of this attack.

4 Modeling and Simulated Attacks

4.1 Unprotected model

As a first step, in order to examine the quality of the modulation technique, an unprotected model has been defined and implemented, then leakage traces were gathered using this model, and finally a simulated analysis attack was performed. An AES-128 encryption algorithm was implemented in code (using Matlab). A constant 128-bit key was set, and known plaintext attacker was assumed (with classification to buckets of subset plaintext bits). The leakage traces were defined as the Hamming Weight (HW) model considering each of the AES rounds. Meaning, the leakage of N plaintext values that inserted to the AES code, is modeled as a matrix of HW values referring to different states inside the AES encryption. Its rows refer to the N inserted plaintexts, i.e., number of traces. Its columns refer to the different ten rounds (to examine the impact of the diffusion characteristic). The analysis attack step examines the modeled leakage traces for a conventional attack and a modulation-based attack using the (conventional) SNR and and SNR_{mod}, respectively. For a proper comparison between these attacks, the exact number of attacked bits (out of the 128-bits) were considered. That is, the number of bits considered for the classification of the internal attacked variable y_i^* in the conventional attack is the same number considered for the observed\classification bits, b, in the modulation-based attack. In addition, these simulated attacks were investigated under various values of standard normally distributed noise added to the modeled leakage traces.

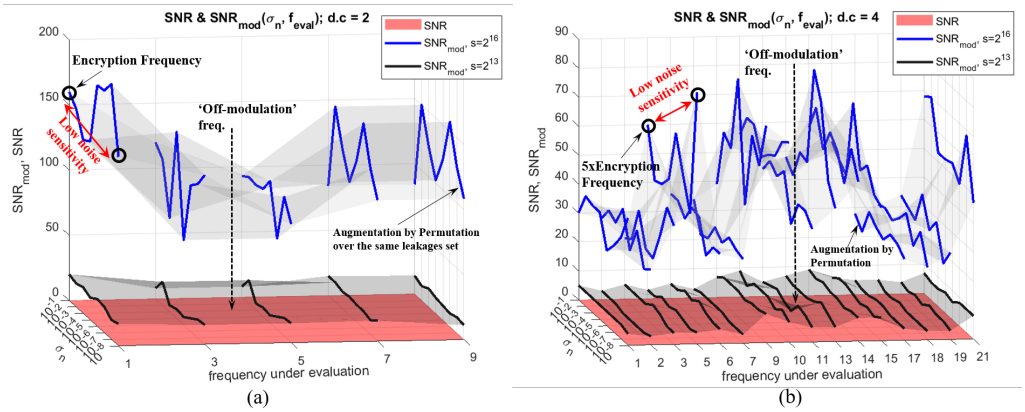
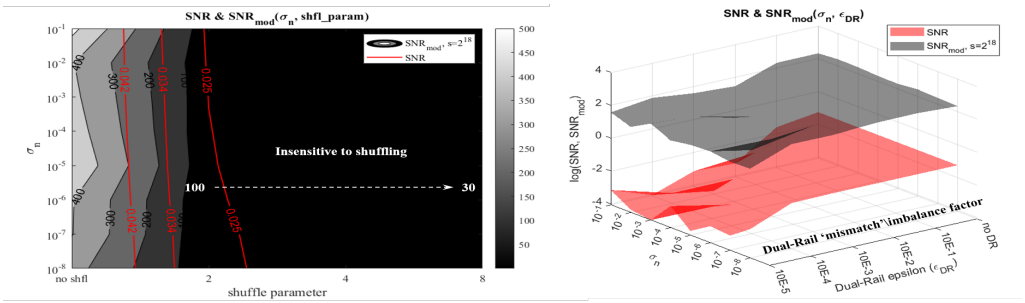


Figure 5: Conventional SNR and SNR_{mod} as a function of noise and frequency under evaluation. (a) d.c=2. (b) d.c=4.

Figure 5 shows an example of the SNR analysis for the conventional and the modulation-based attacks on the HW leakage model. In this example, $N = 10000$ plaintexts were encrypted using the AES-128 algorithm, generating a HW leakage traces for all the ten rounds. The classification part of the SNR calculation for the conventional attack was performed using the three MSB bits of an internal variable y_i^* (after one round), taking into account the entire trace. Similarly, the number of considered classification bits for the modulation-based attack was set to $b = 3$. As can be seen in the figure, the conventional SNR (the surfaces in red) is very low and ranges $10^{-2} - 10^{0}$ for different noise standard-deviation, as shown in the figure. The SNR values slightly depend on the added noise, however since the values are too small, the SNR surfaces seem flattened in this figure scale. On the other hand, the SNR_{mod} values (surfaces in grey and blue) are much higher when calculated on the frequencies that carry information. The same HW leakage traces (produced by the $N = 10000$ plaintexts) were used for analyzing the SNR_{mod}, along with the powerful degrees of freedom parameters, such as the set size s and the duty cycle $d.c$. The grouping parameter g was set to 16. The SNR_{mod} for $d.c = 2$ is shown in Figure 5(a). In this case there are five frequencies under evaluation that carry information, as can be calculated using Equation 5 (where $n = 11$, adding the plaintext to the ten rounds). That is, the information is modulated to the encryption frequency and to four more higher frequencies. As can be seen, when setting the set size parameter to $s = 2^{13}$ (using permutations on parts of the 10000 traces), the SNR_{mod} values increase at least by two orders of magnitude comparing to the conventional SNR. Furthermore, when increasing the set size parameter to $s = 2^{16}$, the SNR_{mod} values significantly rise by around four orders of magnitude comparing to the conventional SNR. Similarly, The SNR_{mod} for $d.c = 4$ is shown in Figure 5(b). This case includes 16 frequencies under evaluation. As expected, The SNR_{mod} values of both set sizes are lower comparing to the SNR_{mod} values of the $d.c = 2$ case. For the same set size, the higher the $d.c$ is, the lower the number of transients between anti-symmetric classes is. Still, these values are much higher than the conventional SNR values. Interesting trade-offs can be noticed through the results of Figure 5: an attacker can increase the number of frequencies carry information (by increasing the $d.c$), on the expense of the obtained SNR_{mod}. Alternatively, an attacker can increase the SNR_{mod} (by increasing s), on the expense of computing time.

4.2 Protected model – Shuffling and Dual-Rail

The second step of examining the modulation technique included an implementation of protected designs with the HW leakage model, based on *Shuffling* and *Dual-Rail*.



(a) (b)

Figure 6: Conventional SNR and SNR_{mod} for protected model. (a) shuffling countermeasure. (b) Dual-Rail countermeasure.

Shuffling: The *shuffling* countermeasure implemented in the HW model is based on randomized execution of the 16 AES-128 S-boxes, depending on a predetermined shuffling parameter, *shfl*. The shuffling implementation in the HW model included the values of $shfl = 2, 4, 8$. This parameter actually determines the number of S-boxes that execute their operation at the same time. As an example, a parameter value of $shfl = 2$ induces an $16/shfl = 8$ randomly chosen S-boxes to execute at the same time, while the rest 8 S-boxes will execute their computation at a different time/cycle. As expected, the shuffling implementation significantly decreases the leaked information when examining the protected model using the conventional SNR, as shown in Figure 6(a). The figure presents a contour diagram of the SNR and SNR_{mod} as a function of the *shfl* parameter and the noise, built using 10000 traces. It can be noticed that the SNR values (red lines) in the contour diagram decrease with the increase of the *shfl* parameter value, with slight dependency on the noise. In contrast, when examining the protected model using the modulation-based technique, the SNR_{mod} distinguisher remains very effective independently on the *shfl* parameter. The SNR_{mod} values (in grey scale), which are around 400 without the shuffling mode, remain very high (above 30) even for a high shuffling parameter values: $shfl = 8$, such that a classification between different observed bits can be easily made by an attacker. Note that the obtained contour diagram in Figure 6(a) relating the the SNR_{mod}, refers to the parameter values of $b = 3, d.c = 2$ and $s = 2^{18}$. As a consequence of the *shuffling* investigation, the modulation-based attack is insensitive to *shuffling* countermeasure, or alternatively said: time-shuffling does not mask the frequency domain characteristic of the leakage when sufficiently long concatenated-traces and permutations are considered.

Dual-Rail: The basic idea of the *dual-rail* approach relies on making the leakage (e.g., power consumption) as much as constant or independent of the processed data. To acheive this, each logical operation is duplicated by using the original operation as well as its complementary operation. This method makes it difficult for an attacker to extract information. However, since the efficiency of the dual-rail technique is based on the symmetry of the gate structure, its immunity in terms of security is sensitive to physical imbalances of the gate's transistors, such as process mismatch, coupling capacitances, process variations, noise, delay imbalance etc. [Tir07, LBBS20]. As a result, the HW model implemented with the *dual-rail* method equals to $(HW_{orig} + \eta) + (HW_{comp} + \eta)$, where the left and right expressions represent the HW (i.e., leakage) of the original and its complimentary operations, respectively, with the addition of noise η. The imbalances between the original and the complementary operations lead to a much smaller amount of leakage, comparing to an unprotected model. That is, the HW leakage model can be assumed as $\epsilon_{DR} \cdot HW_{orig} + 2 \cdot \eta$, where ϵ_{DR} represents the magnitude of the leakage fading. Clearly, the more balanced the design is, the smaller the ϵ_{DR}. Figure 6(b) shows the $log(SNR_{mod})$ and the conventional $log(SNR)$ results of the HW leakage model, as

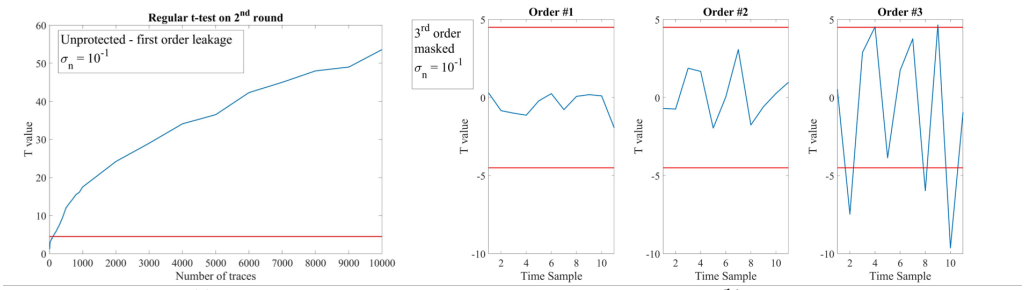


Figure 7: T-test on HW model: (a) Unprotected (1^{st}-Order). (b) 3^{rd}-Order Masked.

a function of the dual-rail epsilon ϵ_{DR} and the noise. As can be clearly noticed, the conventional SNR (red surface) drastically decreases with the decrease of ϵ_{DR}, and with the increase of the noise (which completely blurs the residual leakage). On the other hand, the modulation-based technique results (grey surface), referring to the parameter values of $b = 3, d.c = 2$ and $s = 2^{18}$, show a very moderate decrease in SNR_{mod} along the ϵ_{DR} axis (relative to to the conventional SNR surface). Still, the SNR_{mod} values for the smallest ϵ_{DR} are very high which enables to carry out a classification process of the information.

4.3 Protected model – Masking

As an essential investigation of the proposed modulation-based analysis, the leakage model was implemented while embedding *masking* countermeasure. The masking implementation included the options of $d = 2, 3, 4$ shares[4].

In our model every bit in the AES internal state was masked with $d - 1$ fresh random vectors of size 128-bits, $\mathbf{r_i}$, where the last share was computed using $\mathbf{s} \bigoplus_{i=0}^{i=d-1} \mathbf{r_i}$. In order to evaluate the leaked information, we used the *T-test* metric, considering *Fixed vs. Random* case. Figure 7(a) shows the first-order T-test results in time domain of an unprotected HW model versus the number of the considered traces, with an added standard normal distributed noise of $\sigma_n = 0.1$. As clearly shown, the model leaks information after a small number of traces (around 100 traces). Figure 7(b) shows the T-test results of a masked HW model implementation with $d = 3$ shares as a function of time (which is the AES rounds in the HW model case). As can be noticed, the masked model is completely immune and does not leak information when examined using the first and second orders. The model starts leaking information when examined using the third statistical order only, as expected from theory. However, it requires high computation time and quite huge data-complexity due to a very large number of analyzed traces (around $15M$ traces) that are computed for several orders (statistical moments). In contrast to the time domain, when examining the masked HW model using the modulation-based attack in the frequency domain, the masked leakage becomes much more informative. The combination of the *grouping-interleaving* together with the *F-transforming* operations results in leaked information at several informative frequencies. Even though the encryption is processed on masked data, each informative frequency contains a constellation dependent on an **unmasked** data, which enables an attacker to build an efficient distinguisher from the first statistical moment. The informative masked leakage is analytically explained as follows: Let's simplify the discussion to a single bit resolution (i.e., $b = 1$) and a duty-cycle $d.c = 2$ at the informative encryption frequency f_{enc}, considering n samples for each trace and a set size s. Suppose that $z[n]$ represents the unmasked concatenated trace in time-domain, we will next show that the information obtained at the encryption

[4]Note that d relates to the number of shares corresponding below to information leakage in the d^{th}-order statistical moment, where the security order is $d - 1$

frequency f_{enc} as an output of the $F-transforming$ using the DFT operator can be classified among all the different interleaving possibilities. *This is because different conditional leakage distributions in the secret z are mixing up, which is outside standard masking leakage models (e.g., consider the noisy leakage model).* In general, the DFT operation on a time-domain sequence $z[n]$ is defined as $Z(k) = \frac{1}{M}\sum_{m=0}^{M-1} z[m]e^{\frac{-i2\pi km}{M}}$, where $k = 1, ..., M-1$ represents the frequency domain samples, and M is the number of data samples. Considering the parameters mentioned in this example, $M = s{\cdot}n{\cdot}d.c = 2{\cdot}n{\cdot}s$. Since the encryption occurs every n samples and $d.c = 2$, the corresponding $\frac{k}{M}$ ratio of the encryption frequency f_{enc} equals to $\frac{1}{n}$. As a result, assuming an identity (value) attack model on a masked data $x[m]$ (i.e., $x[m] = r[m] + r[m] \oplus z[m]$, where $r[m]$ is a random sequence and $z[m]$ is the unmasked data), the DFT operation at the encryption frequency is then $X(k_{enc}) = \frac{1}{M}\sum_{m=0}^{M-1}(r[m] + r[m] \oplus z[m])e^{\frac{-i2\pi m}{n}}$. Next we show the different classifications obtained for different interleaving types.

$\mathbf{0 \rightarrow 0}$ interleaving: In this case, the concatenated interleaved (unmasked) trace in time-domain can be presented as $z[n] = \mathbf{0}z_1^0 z_2^0 ... z_{n-1}^0 \mathbf{0} z_{n+1}^0 z_{n+2}^0 ... z_{2n-1}^0 \mathbf{0} z_{2n+1}^0 z_{2n+2}^0 \cdots z_{3n-1}^0 ... \mathbf{0} z_{2{\cdot}n{\cdot}s-n+1}^0 z_{2{\cdot}n{\cdot}s-n+2}^0 ... z_{2{\cdot}n{\cdot}s-1}^0$, where $z_i^0 \in \{0, 1\}$ belong to the 0 classification. Therefore, the *F-transforming* operation of this case using DFT is given in Equation 8.

$$X(k) = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi km}{M}}\Big|_{k=k_{enc}} = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi m}{n}} = \underbrace{\frac{2}{M}\sum_{m=0}^{\frac{M}{n}-1} r[nm]e^{-i2\pi m}}_{\gamma^0} + \underbrace{\frac{1}{M}\sum_{\substack{j=0 \\ j\neq nm}}^{M-1} r[j]e^{\frac{-i2\pi j}{n}}}_{\eta^0} = |\gamma^0 + \eta^0|\underline{\angle\Phi} \quad (8)$$

$\mathbf{1 \rightarrow 1}$ interleaving: In this case, the concatenated interleaved (unmasked) trace in time-domain can be presented as $z[n] = \mathbf{1}z_1^1 z_2^1 ... z_{n-1}^1 \mathbf{1} z_{n+1}^1 z_{n+2}^1 ... z_{2n-1}^1 \mathbf{1} z_{2n+1}^1 z_{2n+2}^1 \cdots z_{3n-1}^1 ... \mathbf{1} z_{2{\cdot}n{\cdot}s-n+1}^1 z_{2{\cdot}n{\cdot}s-n+2}^1 ... z_{2{\cdot}n{\cdot}s-1}^1$, where $z_i^1 \in \{0, 1\}$ belong to the 1 classification. The *F-transforming* operation of this case is given in Equation 9.

$$X(k) = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi km}{M}}\Big|_{k=k_{enc}} = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi m}{n}} = \underbrace{\frac{1}{M}\sum_{m=0}^{\frac{M}{n}-1} e^{-i2\pi m}}_{\approx 0} + \underbrace{\frac{1}{M}\sum_{\substack{j=0 \\ j\neq nm}}^{M-1} r[j]e^{\frac{-i2\pi j}{n}}}_{\eta^1} = |\eta^1|\underline{\angle\Theta} \quad (9)$$

$\mathbf{0 \rightarrow 1}$ interleaving: In this case, the concatenated interleaved (unmasked) trace in time-domain can be presented as $z[n] = \mathbf{0}z_1^0 z_2^0 ... z_{n-1}^0 \mathbf{1} z_{n+1}^1 z_{n+2}^1 ... z_{2n-1}^1 \mathbf{0} z_{2n+1}^0 z_{2n+2}^0 \cdots z_{3n-1}^0 ... \mathbf{1} z_{2{\cdot}n{\cdot}s-n+1}^1 z_{2{\cdot}n{\cdot}s-n+2}^1 ... z_{2{\cdot}n{\cdot}s-1}^1$, where $z_i^0, z_i^1 \in \{0, 1\}$ belong to the 0 and 1 classifications, respectively. The *F-transforming* op. of this case is given in Equation 10.

$$X(k) = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi km}{M}}\Big|_{k=k_{enc}} = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi m}{n}} = \underbrace{\frac{2}{M}\sum_{m=0}^{\frac{M}{2n}-1} r[2nm]e^{-i4\pi m}}_{(\frac{\gamma^0}{2})\underline{\angle\varphi}} \underbrace{\frac{1}{M}\sum_{m=1}^{\frac{M}{2n}} e^{-i2\pi m}}_{\approx 0} + \underbrace{\frac{1}{M}\sum_{\substack{j=0 \\ j\neq nm}}^{M-1} r[j]e^{\frac{-i2\pi j}{n}}}_{(\frac{\eta^0}{2}+\frac{\eta^1}{2})\underline{\angle\phi}} = \left|\frac{\gamma^0+\eta^0+\eta^1}{2}\right|\underline{\angle\psi} \quad (10)$$

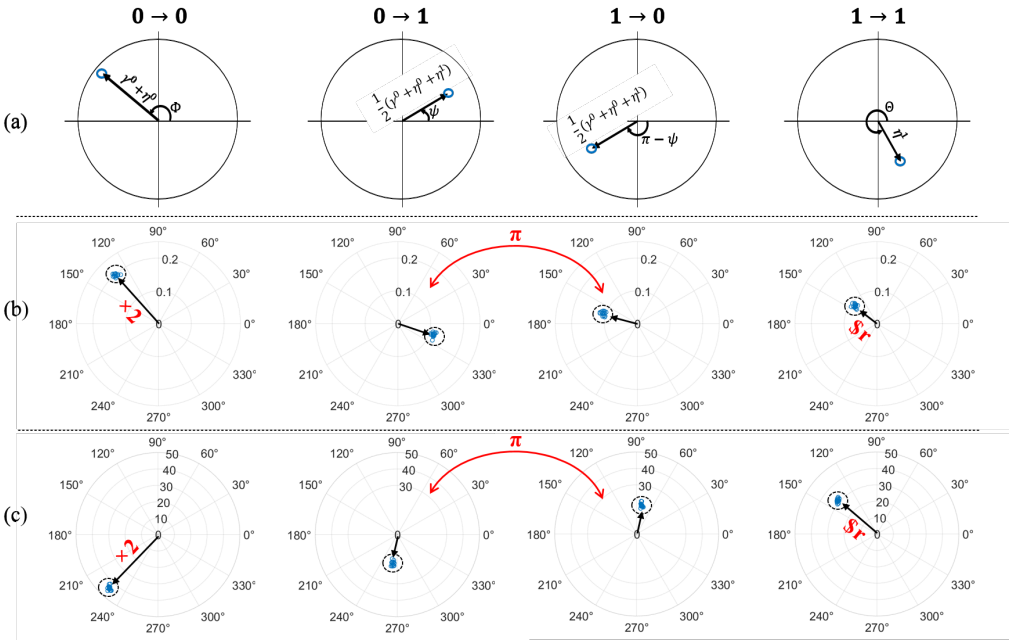


Figure 8: Modulation-based constellations of a masked implementation relating to the four transitions of the analyzed single bit. (a) analytical description referring to Equations 8-9. (b) masking model with 3-shares. (c) masked FPGA with 3-shares.

$1 \rightarrow 0$ interleaving: In this case, the concatenated interleaved (unmasked) trace in time-domain can be presented as $z[n] = \mathbf{1}z_1^1 z_2^1 ... z_{n-1}^1 \mathbf{0} z_{n+1}^0 z_{n+2}^0 ... z_{2n-1}^0 \mathbf{1} z_{2n+1}^1 z_{2n+2}^1 \cdots z_{3n-1}^1 ... \mathbf{0} z_{2\cdot n\cdot s-n+1}^0 z_{2\cdot n\cdot s-n+2}^0 ... z_{2\cdot n\cdot s-1}^0$, where $z_i^0, z_i^1 \in \{0,1\}$ belong to the 0 and 1 classifications, respectively. The *F-transforming* op. of this case is given in Equation 11.

$$X(k) = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi km}{M}}\Big|_{k=k_{enc}} = \frac{1}{M}\sum_{m=0}^{M-1}(r[m]+r[m]\oplus z[m])e^{\frac{-i2\pi m}{n}} =$$
$$\underbrace{\frac{1}{M}\sum_{m=0}^{\frac{M}{2n}-1}e^{-i2\pi m}}_{\approx 0} + \underbrace{\frac{2}{M}\sum_{m=1}^{\frac{M}{2n}}r[2nm]e^{-i4\pi m}}_{(\frac{\gamma^0}{2})\angle\pi-\varphi} + \underbrace{\frac{1}{M}\sum_{\substack{j=0\\ j\neq nm}}^{M-1}r[j]e^{\frac{-i2\pi j}{n}}}_{(\frac{\eta^0}{2}+\frac{\eta^1}{2})\angle\pi-\phi} \approx \left|\frac{\gamma^0+\eta^0+\eta^1}{2}\right|\angle\pi-\psi \tag{11}$$

As can be concluded from these expressions, four contents which differ in their magnitudes and phase are obtained, corresponding to a specific set of sampled traces. Namely, after collecting a sufficient number of traces, the modulation-based analysis on the b bits (of the masked traces) induces a classifiable constellation map in which the magnitudes and phases of the different classes are determined by the unmasked data $z[n]$ and the random sequence of the masking operation $r[n]$. A different collection of traces may induce a different but still a classifiable constellation map that can be utilized by an attacker. An example of a single bit classification (b=1) at the encryption frequency, for duty cycle $d.c = 2$ and set size $s = 2^{22}$ is shown in Figure 8. Note that η^0, γ^0 correspond to the DFT calculation of $r[m], r[nm]$ of the traces that belong to the **0** class. Similarly, η^1 corresponds to the DFT calculation of $r[m]$ of the traces that belong to the **1** class. The different constellation of the analytical expressions is illustrated in Figure 8(a). Accordingly, the constellations for

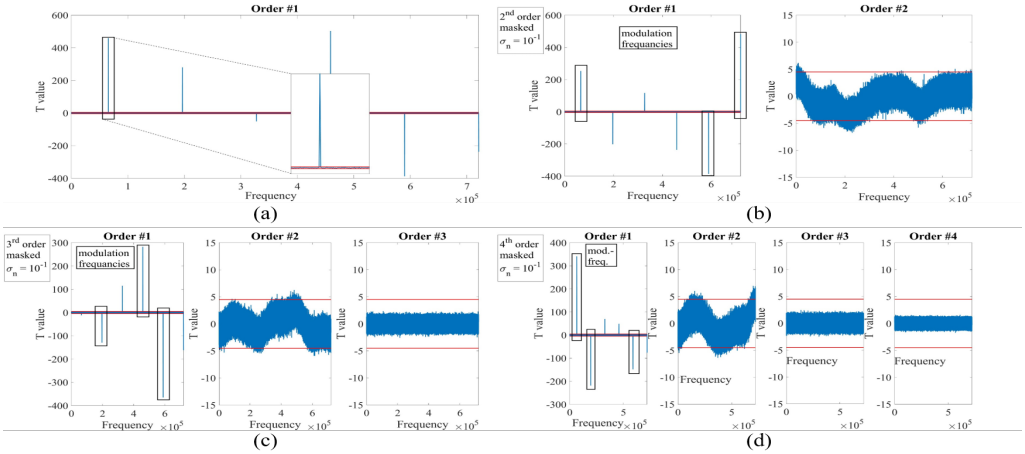


Figure 9: T-test on HW model. (a) unprotected. (b) masking with 2-shares. (c) masking with 3-shares. (d) masking with 4-shares.

the masked leakage model and a masked FPGA implementation[5] analyses are shown in Figures 8(b) and (c), respectively. It is important to note number of insights: the $0 \rightarrow 0$ interleaving is indicated in an approximately doubled magnitude relating to the $0 \rightarrow 1$ and the $1 \rightarrow 0$ cases. The phase difference between the $0 \rightarrow 1$ and the $1 \rightarrow 0$ is π. The $1 \rightarrow 1$ interleaving results in a random magnitude and phase (noted as $\$r$). For any masking order separable constellation maps exist, below we also show FPGA based hardware attack results for $d = 3, 4$. Figure 9 shows T-test results of the masked HW model using the modulation-based analysis, with parameter values of: $b = 3, d.c = 2, s = 2^{16}, \sigma_n = 10^{-1}$. In the case of the modulation-based technique, the T-test is calculated as a function of the frequencies. Similar to the conventional analysis the *Fixed vs. Random* case is considered as well. The *Fixed* traces were defined as the concatenation between traces that belong to the 000 and 111 classes. The *Random* traces were defined as the concatenation between traces from all classes. As expected, the first-order T-test results of an unprotected HW model show significant anomalies in the discrete frequencies that carry information, as shown in Figure 9(a). The T-test results of a masked HW model with two, three and four shares are shown in Figure 9(b),(c),(d), respectively. It can be noticed that the results are consistent with the absolute classification shown in the analytical expression in Equations 8 – 11 and in the illustrations in Figure 8(a). In all of the masked implementations, the models significantly leak information already at the first-order analysis, in the same discrete frequencies. In contrast to the conventional time-domain T-test analysis, the modulation-based analysis is much more efficient since several parameters and assumptions can be determined in advance and reduce the computation time: (1) the POIs in the frequency domain can be defined only as the frequencies that carry information. (2) the set size s can be optimized according to the noise that exists at the analyzed frequency. (3) only the first-order T-test can be examined, and as such, the longer computation time (and data complexity) needed for the second-order and higher orders can be saved.

5 Implementation and Attacks

The next step after showing the modulation-based attack on a HW model, aims to investigate the proposed analysis on a real leakage sampled by physical measurements, first from an unprotected device. For this purpose, an unprotected AES-128 was implemented in software on an STM32F415 model of 32-bit ARM CPU. A chosen plaintext attack was

[5] details on the hardware FPGA implementation are provided below

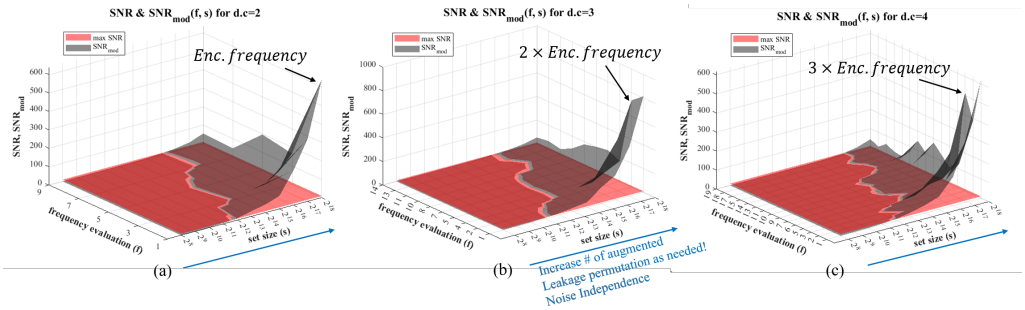


Figure 10: Conventional SNR and SNR_{mod} as a function of set size and frequency under evaluation. (a) d.c=2. (b) d.c=3. (c) d.c=4.

assumed, and power consumption traces of this device were sampled for later analysis, when known random plaintext vectors were encrypted. A conventional SNR and an SNR_{mod} were analyzed using 10240 traces, considering 100 time samples each; a POI was found around the first Sbox output. The SNR and SNR_{mod} results are shown in Figure 10.

The conventional SNR (the red surfaces) was assumed as the maximal obtained SNR value, which was around 9.5 (high SNR for this STM device). The modulation-based SNR_{mod} (the grey surfaces) were calculated considering $b = 3$ observation bits, for various set sizes between 2^8 to 2^{18}, and for various frequencies that carry information. Figures 10(a,b,c) show the results for $d.c = 2, 3, 4$, respectively. It can be noticed that the SNR_{mod} values are smaller than the conventional SNR for a set sizes lower than $s = 2^{12}$. These sizes are too small such that a meaningful information can not be better classified. However, as the set sizes increase, the SNR_{mod} increases as well, much more than the conventional SNR. Furthermore, the information exists at all the frequencies under evaluation, and as was shown, this information can be significantly strengthened by increasing the set size. This powerful fact allows an attacker to control the most convenient frequency band for analysis, extract information with far less traces (orders of magnitude) and much faster.

6 Modeled attacks with countermeasures ON

In this section a hardware based implementation is examined in terms of security evaluation using the conventional and the modulation-based analysis. For this purpose, a masked 4-bit Present algorithm Sbox module was implemented in a Spartan6 FPGA device, with two and three shares. Similar to the previous analysis, a known plaintext attack was assumed with classification abilities, and the power consumption of the device was sampled during the Sbox operations on a standard Sakura-G board utilizing the internal amplifier to a standars Series-5 Picoscope oscilloscope. This masked module was analyzed using the T-test metric as a function of time samples for the conventional attack, and as a function of the frequencies for the proposed modulation-based attack. The results are shown in Figure 11. Similar to the analysis of the masked model shown previously, a *Fixed* vs. *Random* case was considered. For the modulation-based analysis, the *Fixed* traces were defined as the concatenation between traces that belong to the 000 and 111 classes, and the *Random* traces were defined as the concatenation between traces from all classes.

The conventional time-domain T-test was analyzed using $10M$ traces, where considering 100 time samples for each trace for the 2-shares masking measurements (shown in Figure 11(a)), and 10 time samples for the 3-shares masking measurements (Figure 11(c)). Time sample ranges shown are zoomed-in. These results show identifiable information in the second and the third orders for the two and three shares masking implementations, respectively as expacted. However, the leakages were reveled within a high computation time required for the high number of traces as well as high moment calculations. On

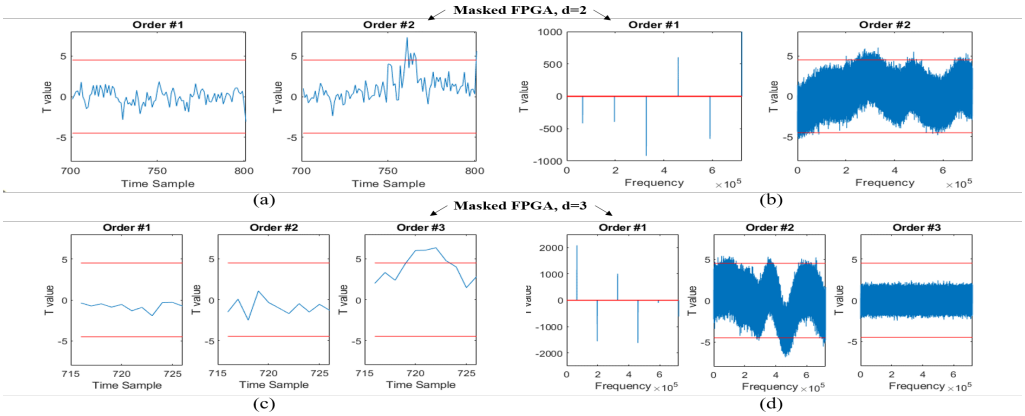


Figure 11: T-test on a masked Sbox implemented in FPGA. (a) conventional T-test on a 2-shares masking. (b) modulation-based T-test on a 2-shares masking. (c) conventional T-test on a 3-shares masking. (d) modulation-based T-test on a 3-shares masking.

the other hand, the modulation-based T-tests for the 2-shares masking (shown in Figure 11(b)) and for the 3-shares masking (Figure 11(d)) were analyzed using **only** 100 **traces**, with parameter values of: $b = 3, d.c = 2, s = 2^{16}$. These results show much higher efficiency than the conventional time-domain analysis, where significant information leakages are obtained in the relevant discrete frequencies already at the first-order moments.

7 Conclusion

In this paper we propose a set of different and powerful techniques for SCA, each provides a contribution by its' own and all work jointly to achieve a low computational cost distinguisher and low data complexity SCA attack. As such, opening the question of how to protect and build sound countermeasures.

We demonstrate security-evaluation of our attack against various leakage models, software implementation and protected designs on an FPGA. All of the discussed techniques are novel leading to a game-change scenario in the SCA context, as discussed above. We demonstrate that by using the proposed *Classification*, *Grouping-Interleaving*, and *F-transforming* techniques an adversary gets access to *shape the characteristics of the physical leakage and its features*. Important aspects are to jointly tailor the leakage to the *spectrum based distinguishers*, and the validity of the transform by-design utilizing long enough concatenated traces and the way they are modulated. One more observation is that leakage is extracted *simultaneously from the entire concatenated long-trace*. Where, computations or internal-variables are jointly correlated to each other in a single leakage (with the limitation of code diffusion characteristics), and this joint-information is modulated periodically by the grouping-interleaving operator to be efficiently captured in the spectrum.

Practically, such an attack changes the way we comprehend security of some countermeasures. It requires re thinking for many countermeasures which we were considering as secure to some level, e.g., masking or shuffling with large permutation size. The following fact simply highlight this point: 100 leakage traces are sufficient to detect leakage from a 3^{rd} order masked design by utilizing the first statistical moment alone as compared to $15 \cdot 10^6$ with conventional third statistical order leakage detection. One possible solution left for future investigation is in trying to build an effective mode-level security such as rekeying, using public tweaks or nonces in a meaningful way to resist the SCMA attack.

Acknowledgments. I. Levi and M. Avital were funded by the Pazy Foundation Research Grant ID377. I. Levi was funded by the Israel Science Foundation (ISF) grant 2569/21.

References

[AASA+20] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. Status report on the second round of the nist post-quantum cryptography standardization process. *US Department of Commerce, NIST*, 2020.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *International workshop on cryptographic hardware and embedded systems*, pages 16–29. Springer, 2004.

[BCPZ16] Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the isw masking scheme. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 23–39. Springer, 2016.

[BFP22] Alessandro Barenghi, Gioele Falcetti, and Gerardo Pelosi. Locating side channel leakage in time through matched filters. *Cryptography*, 6(2):26, 2022.

[BGH+15] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Less is more: dimensionality reduction from a theoretical perspective. In *Cryptographic Hardware and Embedded Systems–CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings 17*, pages 22–41. Springer, 2015.

[BGH+17] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, Damien Marion, and Olivier Rioul. Optimal side-channel attacks for multivariate leakages and multiple models. *Journal of Cryptographic Engineering*, 7:331–341, 2017.

[Bog07] Andrey Bogdanov. Improved side-channel collision attacks on aes. In *International Workshop on Selected Areas in Cryptography*, pages 84–95. Springer, 2007.

[CDP16] Eleonora Cagli, Cécile Dumas, and Emmanuel Prouff. Enhancing dimensionality reduction methods for side-channel attacks. In *Smart Card Research and Advanced Applications: 14th International Conference, CARDIS 2015, Bochum, Germany, November 4-6, 2015. Revised Selected Papers 14*, pages 15–33. Springer, 2016.

[CGLS20] Gaëtan Cassiers, Benjamin Grégoire, Itamar Levi, and François-Xavier Standaert. Hardware private circuits: From trivial composition to full verification. *IEEE Transactions on Computers*, 2020.

[CJRR99] Suresh Chari, Charanjit S Jutla, Josyula R Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *Advances in Cryptology—CRYPTO'99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings 19*, pages 398–412. Springer, 1999.

[CMG+] Jeremy Cooper, Elke De Mulder, Gilbert Goodwill, Josh Jaffe, Gary Kenworthy, and Pankaj Rohatgi. Test vector leakage assessment (TVLA) methodology in practice (extended abstract). ICMC 2013.

[CPRR14] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In *Fast Software Encryption: 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers*, pages 410–424. Springer, 2014.

[CRR02] Suresh Chari, Josyula R Rao, and Pankaj Rohatgi. Template attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 13–28. Springer, 2002.

[DPRS11] Julien Doget, Emmanuel Prouff, Matthieu Rivain, and François-Xavier Standaert. Univariate side channel attacks and leakage modeling. *Cryptology ePrint Archive*, 2011.

[DS16] François Durvaux and François-Xavier Standaert. From improved leakage detection to the detection of points of interests in leakage traces. In *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I 35*, pages 240–262. Springer, 2016.

[DSEA+12] Nicolas Debande, Youssef Souissi, M Abdelaziz El Aabid, Sylvain Guilley, and Jean-Luc Danger. Wavelet transform based pre-processing for side channel analysis. In *2012 45th Annual IEEE/ACM International Symposium on Microarchitecture Workshops*, pages 32–38. IEEE, 2012.

[FMPR11] Guillaume Fumaroli, Ange Martinelli, Emmanuel Prouff, and Matthieu Rivain. Affine masking against higher-order side channel analysis. In *Selected Areas in Cryptography: 17th International Workshop, SAC 2010, Waterloo, Ontario, Canada, August 12-13, 2010, Revised Selected Papers 17*, pages 262–280. Springer, 2011.

[GGJR+11] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In *Annual International Cryptology Conference*, pages 463–481. Springer, 2003.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Annual international cryptology conference*, pages 388–397. Springer, 1999.

[LBBS20] Itamar Levi, Davide Bellizia, David Bol, and François-Xavier Standaert. Ask less, get more: Side-channel signal hiding, revisited. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2020.

[LBS20] Itamar Levi, Davide Bellizia, and François-Xavier Standaert. Beyond algorithmic noise or how to shuffle parallel implementations? *International Journal of Circuit Theory and Applications*, 48(5):674–695, 2020.

[LCS+21] Bozhi Liu, Kemeng Chen, Minjun Seo, Janet M Roveda, and Roman Lysecky. Methods and analysis of automated trace alignment under power obfuscation in side channel attacks. *Journal of Hardware and Systems Security*, 5(2):127–142, 2021.

[LH20] JongHyeok Lee and Dong-Guk Han. Security analysis on dummy based side-channel countermeasures—case study: Aes with dummy and shuffling. *Applied Soft Computing*, 93:106352, 2020.

[LKF15] Itamar Levi, Osnat Keren, and Alexander Fish. Data-dependent delays as a barrier against power attacks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(8):2069–2078, 2015.

[LPB+15] Liran Lerman, Romain Poussier, Gianluca Bontempi, Olivier Markowitch, and François-Xavier Standaert. Template attacks vs. machine learning revisited (and the curse of dimensionality in side-channel analysis). In *International Workshop on Constructive Side-Channel Analysis and Secure Design*, pages 20–33. Springer, 2015.

[LPR+14] Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In *Cryptographic Hardware and Embedded Systems–CHES 2014: 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings 16*, pages 35–54. Springer, 2014.

[Man04] Stefan Mangard. Hardware countermeasures against dpa–a statistical analysis of their effectiveness. In *Cryptographers' Track at the RSA Conference*, pages 222–235. Springer, 2004.

[MG10] Edgar Mateos and Catherine H Gebotys. A new correlation frequency analysis of the side channel. In *Proceedings of the 5th Workshop on Embedded Systems Security*, pages 1–8, 2010.

[MS16] Amir Moradi and François-Xavier Standaert. Moments-correlating dpa. In *Proceedings of the 2016 ACM Workshop on Theory of Implementation Security*, pages 5–15, 2016.

[MvWB11] Ruben A Muijrers, Jasper GJ van Woudenberg, and Lejla Batina. Ram: Rapid alignment method. In *Smart Card Research and Advanced Applications: 10th IFIP WG 8.8/11.2 International Conference, CARDIS 2011, Leuven, Belgium, September 14-16, 2011, Revised Selected Papers 10*, pages 266–282. Springer, 2011.

[OM93] Ralph G O'Brien and Keith E Muller. Unified power analysis for t-tests through multivariate hypotheses. *Applied analysis of variance in behavioral science*, pages 297–344, 1993.

[PHF08] Thomas Plos, Michael Hutter, and Martin Feldhofer. Evaluation of side-channel preprocessing techniques on cryptographic-enabled hf and uhf rfid-tag prototypes. In *Workshop on RFID Security*, pages 114–127, 2008.

[Riv08] Matthieu Rivain. On the exact success rate of side channel analysis in the gaussian model. In *Selected Areas in Cryptography*, volume 5381, pages 165–183. Springer, 2008.

[SKS09] François-Xavier Standaert, François Koeune, and Werner Schindler. How to compare profiled side-channel attacks? In *International Conference on Applied Cryptography and Network Security*, pages 485–498. Springer, 2009.

[SLFP04] Kai Schramm, Gregor Leander, Patrick Felke, and Christof Paar. A collision-attack on aes. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 163–175. Springer, 2004.

[SM15] Tobias Schneider and Amir Moradi. Leakage assessment methodology: A clear roadmap for side-channel evaluations. In *Cryptographic Hardware and Embedded Systems–CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings 17*, pages 495–513. Springer, 2015.

[SMY09] François-Xavier Standaert, Tal G Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 443–461. Springer, 2009.

[Sta19] François-Xavier Standaert. How (not) to use welch's t-test in side-channel security evaluations. In *Smart Card Research and Advanced Applications: 17th International Conference, CARDIS 2018, Montpellier, France, November 12–14, 2018, Revised Selected Papers 17*, pages 65–79. Springer, 2019.

[SWL21] Dor Salomon, Amir Weiss, and Itamar Levi. Improved filtering techniques for single-and multi-trace side-channel analysis. *Cryptography*, 5(3):24, 2021.

[SWP03] Kai Schramm, Thomas Wollinger, and Christof Paar. A new class of collision attacks and its application to des. In *International Workshop on Fast Software Encryption*, pages 206–222. Springer, 2003.

[TGWC18] Hugues Thiebeauld, Georges Gagnerot, Antoine Wurcker, and Christophe Clavier. Scatter: A new dimension in side-channel. In *Constructive Side-Channel Analysis and Secure Design: 9th International Workshop, COSADE 2018, Singapore, April 23–24, 2018, Proceedings 9*, pages 135–152. Springer, 2018.

[Tir07] Kris Tiri. Side-channel attack pitfalls. In *Proceedings of the 44th annual Design Automation Conference*, pages 15–20, 2007.

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

[TMC+21] Meltem Sönmez Turan, Kerry McKay, Donghoon Chang, Cagdas Calik, Lawrence Bassham, Jinkeon Kang, John Kelsey, et al. Status report on the second round of the nist lightweight cryptography standardization process. *National Institute of Standards and Technology Internal Report*, 8369(10.6028), 2021.

[TV04] Kris Tiri and Ingrid Verbauwhede. A logic level design methodology for a secure dpa resistant asic or fpga implementation. In *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, volume 1, pages 246–251. IEEE, 2004.

[VCMKS12] Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 740–757. Springer, 2012.

[VCS10] Nicolas Veyrat-Charvillon and François-Xavier Standaert. Adaptive chosen-message side-channel attacks. In *International Conference on Applied Cryptography and Network Security*, pages 186–199. Springer, 2010.

[vWWB11] Jasper GJ van Woudenberg, Marc F Witteman, and Bram Bakker. Improving differential power analysis by elastic alignment. In *Cryptographers' Track at the RSA Conference*, pages 104–119. Springer, 2011.

[Wel47] Bernard L Welch. The generalization of 'students' problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, 1947.

[YWQ09] Lin Yang, Meiqin Wang, and Siyuan Qiao. Side channel cube attack on present. In *International Conference on Cryptology and Network Security*, pages 379–391. Springer, 2009.

[ZWG11] XinJie Zhao, Tao Wang, and ShiZe Guo. Improved side channel cube attacks on present. *Cryptology ePrint Archive*, 2011.