A Unified Formalism for Physical Attacks

Hélène Le Bouder¹,², Ronan Lashermes¹,³, Yanis Linge⁴, Bruno Robisson¹,³, and Assia Tria¹,³

¹ Département Systèmes et Architectures Sécurisés
² École Nationale Supérieure des Mines de Saint-Étienne
³ CEA-Tech
⁴ STMicroelectronics

Abstract. The security of cryptographic algorithms can be considered in two contexts. On the one hand, these algorithms can be proven secure mathematically. On the other hand, physical attacks can weaken the implementation of an algorithm yet proven secure. Under the common name of physical attacks, different attacks are regrouped: side channel attacks and fault injection attacks. This paper presents a common formalism for these attacks and highlights their underlying principles. All physical attacks on symmetric algorithms can be described with a 3-step process. Moreover it is possible to compare different physical attacks, by separating the theoretical attack path and the experimental parts of the attacks.

Keywords: Formalism, physical attacks, side channels, fault injections

1 Introduction

When discussing about the security of a cryptographic algorithm, numerous tools allow the cryptographers to prove the security of a cipher. Unfortunately those tools do not consider the interaction of the computing unit with its physical environment. Physical attacks are a real threat, even for cryptographic algorithms proved secure mathematically. Physical attacks are divided in two families: the side channel attacks and the fault injection attacks and can target the secret key or reverse engineer the algorithm. Regrettably, no framework exists that gathers physical attacks under a common formalism. Our paper proposes one for secret key ciphers and presents a new tool to compare them.

The paper is organised as follows. In the section 2 we expose the context and the motivations for a unifying formalism. A 3-step description of physical attacks is given in section 3. How to compare these attacks is explained in section 4. Some examples of the utilization of the formalism are exposed in 5. Finally the conclusion is drawn in section 6.
2 Context

2.1 Notations and Algorithm

Our formalism is valid for secret key ciphers only. The well known algorithm AES [1] has been chosen as an example in order to illustrate it. In this article, the plaintext is noted \( T \), the ciphertext is noted \( C \), the cipher key is noted \( K \).

**Advanced Encryption Standard** The AES is a standard established by the NIST [1] for symmetric key cryptography. In this paper, we focus on the 128-bits key version of the AES. It is a block cipher, as illustrated in Fig. 1.

\[ K_r \] is the derived key used at round \( r \). The encryption first consists in mapping the plaintext \( T \) of 128 bits into a two-dimensional array of bytes, called the State. Then, after a preliminary XOR between the input and the key, \( AES_{128} \) executes 10 times a function that operates on the State. The operations used during these rounds are:
- **SubBytes** is composed of non-linear transformations: 16 s-boxes noted $SB$, working independently on individual bytes of the State.
- **ShiftRows** noted $SR$, a byte shifting operation on each row of the State.
- **MixColumns** noted $MC$, a linear matrix multiplication working on each column of the State.
- **AddRoundKey** noted $ARK$, a byte-wise xor $\oplus$ between the State and $K_r$, $r \in [0, 10]$.

### 2.2 Physical attacks

A physical access to the circuit may give information about the internal computations of the cryptographic algorithm and thus, indirectly, about secret values in the algorithm. We distinguish physical attacks according to two criteria: their principle and their goal.

- There are two main kinds of physical attacks, the side channel attacks and the fault injection attacks.
  - **Side Channel Attacks** noted SCA are based on observations of the circuit behaviour during the computation, for example in [2,3]. Side channel attacks exploit the fact that some physical values of a circuit depend on intermediary values of the computation. This is the so-called leakage of information of the circuit.

  - **Fault Injection Attacks** notes FIA consist in disturbing the circuit behaviour in order to alter the correct progress of the algorithm [4]. Faults are injected into the device using various means such as laser, clock glitches, spikes on the power supply or electromagnetic perturbations. A rougher technique consists in modifying the circuit operation by modifying internal computations through micro-probes, or even modifying the circuit itself using Focused Ion Beam. In this paper a faulty variable is denoted by an asterisk $^*$. The current trend is to mix the two kinds in order to build other attacks. These are called hybrid attacks, as for example in [5,6].

  - The target or the secret noted $K$ is the goal of the attack. The domain of definition of the target is noted $K$. The target is often the cipher key $K$, but physical attacks are also used to reverse engineer an algorithm, i.e. the goal is to retrieve information on a private algorithm. An example may be an AES with custom and secret s-boxes: in this case, the attacker knows the key and he tries to recover the s-boxes as in section 5.6.

### 2.3 Motivations behind our formalism

Physical attacks add another dimension with respect to cryptanalysis: the interaction of the implemented algorithm with the physical environment. This
work intends to be a guide to help identify the important parameters in a physical attack, so that they can be compared. Unifying the different attacks under a same formalism allows to deal with them with common tools.

Several works have been proposed to described them with a common framework [7,8,9,10,11,12,13,14]. However these works only cover side channel attacks. In particular, [8,14] propose to write various side channel attacks all as Differential Power Analyses. Likewise there are frameworks for fault injection attacks [15,16,17]. The improvement of our formalism is to unify the two families. As a consequence, it becomes possible to compare side channel attacks and fault injection attacks. In [11], Standaert et al. underline the interface between theory and practice for side channel attacks, we enlarge this vision for both families. Finally we want to compare the attacks between them in order to guide the designer in the choices required to harden its implementation.

3 Description of the attacks in 3-step by the formalism

All attack on symmetric ciphers can be decomposed in the following 3-step process.

3.1 Step 1: Campaign

An experiment $E$ is a pair $(O_S, O_R)$ of measurements called observables, taken during the execution of the cipher algorithm of one text $T$ with a key $K$. $O_S$ is called a stimuli and $O_R$ a reaction.

A set of $N$ experiments is called a campaign. Examples of observables are: plaintext, ciphertext, faulty ciphertext, EM traces, power traces, signal provided by a micro-probe, behaviour (i.e. the normal or abnormal working) of the circuit or computation time.

A classical example of pairs of observables for a SCA is

$$(O_S = T, O_R = pow)$$

A classical example of pairs of observables for a FIA is

$$(O_S = C, O_R = C^*)$$

The attack path is an exploitable relation $R$ between our observables and the target $K$. Inputs are $K$ and $O_S$ and the output is $O_R$. This relation is composed of physical functions $f_j$ and algorithm functions $g_i$.

$$O_R = R(O_S, K)$$

$$R = f_1 \circ g_1 \circ \cdots \circ f_n \circ g_n \quad (1)$$

More precisely the $g_j$ are composed of several cipher-algorithm functions. For example in the case of AES, $g_j$ is built with $\{SB, SR, MC, ARK\}$.

The physical functions $f_j$ cannot always be described with a mathematical expression since they are often non deterministic. There is often only one physical function.
3.2 Step 2: Predictions

In the attack path $R$ there are two unknowns: the target $K$ and the physical functions $f$. To build predictions they have to be replaced, either by guesses or models.

The attacker make guesses $k$ on the target $K$. The good guess is noted $\hat{k}$. A divide and conquer approach is generally chosen. The domain of definition of the target, $K$, should be short enough so that all guesses can be tested.

For example if the target is the cipher key of an AES. Generally a round key $K_r$ is chosen as a target and is split in bytes. So $K = [0, 255]$ and has size 256.

As already pointed out, physical functions $f$ do not always have a mathematical expression. But $f$ can be approximated by mathematical functions $m$ called models. In fault injection attacks, models are called error functions and leakage functions in side channel attacks. It is impossible to test every possible model because of their tremendous number. Indeed each endomorphism of a set of bits can be considered as a possible error function, so the number of these functions is too big to be efficiently managed. Even worse, the number of leakage functions is an infinite uncountable set. Commonly, one or a small set of models is used.

Finally the predictions are built with the attack path described in Step 1 (section 3.1) for each guess $h$ on the secret, where the physical functions are replaced by models $m$.

1. Define $K$, the set of guesses $k$ on the target $K$.
2. Choose one set of models $m = \{m_j\}$ with $j \in [1,n]$ where $n$ is the number of physical functions in $R$. In practice, several models $m_{i,j}$ can be tested for one physical function $f_j$. There are two possibilities. The first is to use the models separately, for example in SCA, the attacker often has the choice between Hamming weight or Hamming distance as a model of the power consumption. The second is to take into account all models at once. As an example, it is possible to choose a single-bit fault in a FIA, where there are 8 models for one byte $m_i = 2^i$ with $i \in [0,7]$.
3. Replace the physical functions $f_j$ by models $m_j$ in the equation Eq. (1).
4. For each $h$, and each set of models $m$, compute $P_{m,k}$ with the stimuli $O_S$.

$$P_{m,k} = R_m(O_S, h)$$
$$R_m = m_1 \circ g_1 \circ \cdots \circ m_n \circ g_n \tag{2}$$

3.3 Step 3: Confrontation

For each hypothesis $k$ and set of models $m$, $P_{m,k}$ is confronted with respect to $O_R$ with a distinguisher. A distinguisher is a statistic tool which is able to find the correct guess on the target. The distinguishers highlight links between physical functions $f$ and mathematical models $m$, they are based on different statistical criteria. $P_{m,k}$ and $O_R$ can be considered as random variables. The
Some examples of distinguishers are listed in the following paragraph.

- Sieves and counters [4] are used essentially for fault injection attacks. They suppose that models $m$ are identical to the physical functions $f$.
- The difference of mean [2] and the correlation [3] are based on a linear dependency between the model and the physical function.
- The entropy [18] and the mutual information [19] are based on the Shannon information and use non-linear dependencies between the model and the physical function.
- The Kolmogorov-Smirnov test [20] is a seemingly attractive alternative to mutual information, it is similarly able to generically compare the distributions of two samples but achieves this without explicit estimation of their probability density functions.
- The differential cluster analysis [21] analyses the variances.
- The principal component [22] and the linear discriminant [23] use properties of inter-class and intra-class variances.

The difference and the comparisons of these distinguishers are explained in [24]. There is an active ongoing research on the distinguishers but they are out of the scope of this paper.

3.4 Attack step by step

Some attacks cannot directly retrieve the target. They first require to retrieve one (or several) intermediate secret(s). In this case the 3-step approach described in 3.1, 3.2 and 3.3 is repeated many times, for different attack paths with different targets and observables. We note as an exponent the order of the attack path. The target $K^i$ is an observable for the attack path $R^{i+1}$.

$$R^{i+1}(O_S^{i+1}, K^{i+1}) = O_R^{i+1} \text{ with } K^i = O_S^{i+1} \text{ or } O_R^{i+1}$$
4 How to compare physical attacks?

In this section a compare tool for all physical attacks is presented.

4.1 State of the art

The question that naturally arises is: how to compare physical attacks? They are often compared in terms of equipments: how much they cost, what kind of fault injections are needed. They are also compared by the computational power of the attack they require. The number of measurements needed or the ability to modify the circuit are also points of comparison. For example the different kinds of fault injection possible for a given equipment are described in [25]. In the case of side channel attacks, a lot of work has been done to compare distinguishers as in [24]. Generally we compare different distinguishers on a same circuit or a same distinguisher on different circuits or different models for a same distinguisher. Indeed, as said in section 3.3 different distinguishers do not rely on the same statistical properties.

This paper presents a different approach. The main idea is to evaluate a physical attack in the two first steps. First in a theoretical study, the models are evaluated independently from the physical functions \( i.e \ R_m \) in equation Eq. (2) is studied. Only in a second time the uncertainty due to the use of experimental apparatus is taken into account. The adequacy of the models with respect to the physical functions is evaluated, but without distinguishers, since the comparison of physical attacks is performed before the step of confrontation 3.3.

4.2 Oracle

In the second step 3.2 of our formalism, the predictions are computed. The set of predictions noted \( P \), has a cardinal \( \alpha \) (i.e. there are \( \alpha \) predictions possible). One has to remark that it is possible that \( \alpha \neq \text{card}(K) \). Indeed two guesses can have the same prediction during an attack.

\[
\begin{align*}
O_S & \quad \Theta_m \quad P_{m,k}
\end{align*}
\]

Fig. 2: Oracle \( \Theta_m \)

Let \( \Theta_m \) be an oracle associated with a set of models \( m \) as illustrated in Fig. 2. The oracle \( \Theta_m \) returns \( P_{m,k} = R_m(O_S, K) \) i.e the prediction which corresponds to the good guess under models \( m \). The required number of queries (in average) to \( \Theta_m \) in order to retrieve the target \( K \) is noted \( \theta \). It is a measure of how efficient the attack path would be with the set of models \( m \), i.e \( R_m \).
An oracle can combine sets of models. Let \( m \) be a set of (set of) models \( m_i \). For each \( m_i \) there is an oracle \( \Theta_{m_i} \). An oracle \( \Theta_m \) can be constructed by randomly choosing a model \( m_i \) at each call and returning the result of \( \Theta_{m_i} \).

There is often only one physical function used in a relation \( R \) and therefore only one model in the set \( m \) used in \( R \). Yet, several possibilities are evaluated (e.g. 8 fault models to approximate a single-bit fault model) which form the set \( m \) in our previous example,

\[
m_i = 2^i \quad i \in [0, 7]
\]

### 4.3 Matching Probability

This section deals with the link between observables \( O_R \) and the good prediction \( P_{m,k} \). More precisely the two codomains, for the attack path \( R \) and for \( R_m \) are compared (for a given target \( K \) and a given set of models \( m \)). One has to remark that the attack path \( R \) is not really a function since for a same input \( O_S \), different outputs \( O_R \) are possible. Additionally there is not necessarily a bijection between the codomains (the prediction \( P_{m,k} \) and the observables \( O_R \)), as it has already been shown in [7,10,11,14].

A contingency table is filled with the results of \( N \) experiments. All the possible values of \( P_{m,k} \) are noted \( \rho_i, i \in [1, \alpha] \) and the possible values of \( O_R \) are noted \( o_j, j \in [1, \beta] \). For each experiment \( E = (O_S, O_R) \), the reaction \( O_R \) is stored and \( P_{m,k} \) is computed. Then in the contingency table, shown in Table 1, the value at the corresponding row \( i \) is incremented (prediction is equal to \( \rho_i \)) and column \( j \) (reaction is equal to \( o_j \)). At the end, the value \( a_{i,j} \) is the number of times the attacker computed the prediction \( \rho_i \) in conjunction with the measurement of the reaction \( o_j \), i.e. the number of experiments which verifies \( P_{m,k} = \rho_i \) and \( O_R = o_j \).

<table>
<thead>
<tr>
<th>( P_{m,k} )</th>
<th>( O_R = o_1 )</th>
<th>( O_R = o_2 )</th>
<th>( \cdots )</th>
<th>( O_R = o_\beta )</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{m,k} = \rho_1 )</td>
<td>( a_{1,1} )</td>
<td>( a_{1,2} )</td>
<td>( \cdots )</td>
<td>( a_{1,\beta} )</td>
<td>( \sum_{i=1}^{\beta} a_{1,j} )</td>
</tr>
<tr>
<td>( P_{m,k} = \rho_2 )</td>
<td>( a_{2,1} )</td>
<td>( a_{2,2} )</td>
<td>( \cdots )</td>
<td>( a_{2,\beta} )</td>
<td>( \sum_{i=1}^{\beta} a_{2,j} )</td>
</tr>
<tr>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
<td>( \vdots )</td>
</tr>
<tr>
<td>( P_{m,k} = \rho_\alpha )</td>
<td>( a_{\alpha,1} )</td>
<td>( a_{\alpha,2} )</td>
<td>( \cdots )</td>
<td>( a_{\alpha,\beta} )</td>
<td>( \sum_{i=1}^{\beta} a_{\alpha,j} )</td>
</tr>
<tr>
<td>total</td>
<td>( \sum_{i=1}^{\beta} a_{i,1} )</td>
<td>( \sum_{i=1}^{\beta} a_{i,2} )</td>
<td>( \cdots )</td>
<td>( \sum_{i=1}^{\beta} a_{i,\beta} )</td>
<td>( N )</td>
</tr>
</tbody>
</table>

Table 1: Contingency table of the measured \( O_R \) and the predicted values \( P_{m,k} \)

We now focus on the repeatability probability. An experience \( E \) is associated with a pair \( (O_R = \hat{o}_j, P_{m,k} = \hat{\rho} = \rho_i) \). What is the probability that another experiment with the same observable has also the same predictive value? This probability is noted \( p \).

In order to estimate \( p \), a statistical experience using the Table 1 is presented. A first urn contains a total of \( N \) balls, specifically for each \( i, j, a_{i,j} \) balls are
labelled with the pair \((\rho_i, o_j)\). It represents the Table 1. In the other hand, we have \(\beta\) urns. With urn \(j\) containing \(a_{i,j}\) balls labelled with \(\rho_i\) for each \(i\). Each urn represents a column of Table 1. The statistical experience is the following:

1. A random draw in the first urn. The result is \((\hat{o} = o_{j_1}, \hat{\rho} = \rho_{i_1})\).
2. A random draw in the urn \(j_1\) (\(j_1\) defined by the first draw). The result is \(\rho_{i_2}\).

We are interested in the probability:

\[p = P(\rho_{i_2} = \rho_{i_1})\]

\[P(\rho_{i_2} = \rho_{i_1} | o_{j_1}) = \sum_{i=1}^{N} \frac{a_{i,j}^2}{n^2}\]

Finally we have:

\[p = \sum_{j=1}^{M} P(\rho_{i_2} = \rho_{i_1} | o_{j_2}) \cdot P(o_{j_2}) = \sum_{j=1}^{M} \left( \frac{o_{j}}{n} \cdot \sum_{i=1}^{N} \frac{a_{i,j}^2}{n^2} \right)\]

This probability \(p\) is called matching probability.

### 4.4 A tool to compare physical attacks

The oracle \(\theta\) allows to evaluate the quality of an attack path \(R_m\) with the models \(m\). A smaller \(\theta\) means a better adequacy between the attack path and the model. The matching probability \(p\) represents the quality of the measures \(O_R\) with respect to the predictions \(P_{m,k}\). A bigger \(p\) means a better adequacy between the physical function and the model. Finally \(\theta\) and \(p\) are combined to globally evaluate the experimental attack with respect to the models.

A new oracle \(\Theta_p\) (Fig.3) is introduced which mixes \(\Theta_m\) and \(p\). It returns \(P_{m,k}\) thank to \(O_S\) but with a probability of success \(p\). For \(n\) experimentations, the probability to have \(\theta\) successes is computed with a Bernoulli trial. The probability \(P\) of \(\theta\) or more successes in the experiment \(B(n, p)\) is given by the equation Eq. (4):

\[P = P(\text{obtain at least } \theta \text{ success}) = \sum_{\theta} \binom{n}{\theta} p^\theta \cdot (1-p)^{n-\theta}\]

Finally the new tool to compare the physical attacks is \(P\) for \(n\) texts. This probability can be seen as a success rate. \(P\) depends on the models and the measurements. Another possibility is to compute the number of experiments \(n\) needed to have \(P > 0.99\).
The best attack has the biggest $P$ for a given $n$.
Or the best attack has the smallest $n$ for $P > 0.99$.

The improvements of this probability of success $P$ come from the fact that it is decomposed in two parts. So if an attack fails, it is possible to know if the model is wrong (too different w.r.t. the reality) or if the attack path is bad.

The third step of the formalism is not used in this comparison tool. However the choice of the distinguisher is important for an attack to succeed. In this paper, the idea is to compare attacks before this step.
5 Examples

To illustrate our formalism description, some classic attacks on AES are chosen. Three attacks which aim at finding a cipher key were realized on the same programmable circuit: FPGA - Xilinx Spartan3 700A. The same cipher key A3B0D09804269DC7FB0CDABAD57F8 and the same 5000 plaintexts were used.

5.1 Experimental protocol

In this section, the description of the acquisition benches is given.

**EM bench**  The EM curves are acquired with the protocols described by Dehbaoui in [26]. The EM bench is composed of a control computer, an oscilloscope, and a commercial EM probe (with large bandwidth). At the contact of the circuit, the EM probe receives the EM emission, then sends the measurements to the oscilloscope. A trigger signal sent by the FPGA indicates when the oscilloscope saves the traces. The experimental protocol is illustrated in Fig. 4.

![Fig. 4: Scheme of EM bench](image-url)
Clock glitch bench  The clock glitch bench is described in details by Zussa et al. in [27,28]. The idea is to change the internal clock and to reduce the frequency of the clock to obtain faults. The glitchy clock was generated using a Xilinx Virtex-5 FPGA, used as a clock source for our targeted FPGA.

![Scheme of clock glitch bench](image)

Fig. 5: Scheme of clock glitch bench

5.2 Example of a Side Channel Attack (SCA) on AES

The first attack presented is a classic attack by electromagnetic analysis on the last round of the AES.

- The target is a byte of the key at round 10, $K = K_{10}$. There are only 256 possible values.
- The stimuli is the ciphertext $O_S = C$.
- The reaction is the measured electromagnetic field, $O_R = EM$ curve.
- $R = f \circ g$ with $f = em$ electromagnetic radiation and $g = SB^{-1} \circ SR^{-1} \circ \oplus$.

The attack path is illustrated in Fig. 6

- $R_m = m \circ g$ with $m = HD$ Hamming distance.
So $P_{HD,h} = HD(C, SB^{-1} \circ SR^{-1} \circ (C \oplus h))$
- The distinguisher can be a difference of mean [2], a correlation [3], a mutual information [19], a principal component [22] or a linear discriminant [23].
- The question to evaluate $\theta$ is how many pairs $(C, P_{HD,K_{10}})$ are needed in average to recover a byte of $K_{10}$? In this case we have $\theta = 4$. This value
was obtained by simulating 1,000,000 attacks using the Hamming distance as an input.

- For the different bytes of the round key $K_{10}$, the contingency table is generated to compute the matching probability.

The $O_R$ are the measured values at the instant of the EM curves were a correlation allows to find the value of the target and the $P_{m,K}$ are Hamming distances. The attack has not succeeded for the bytes 3 and 13. Finally, the results are displayed in Table 2.

<table>
<thead>
<tr>
<th>byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>0.22</td>
<td>0.22</td>
<td>0.21</td>
<td>×</td>
<td>0.22</td>
<td>0.22</td>
<td>0.21</td>
<td>0.21</td>
<td>0.20</td>
<td>0.20</td>
<td>0.21</td>
<td>0.21</td>
<td>0.23</td>
<td>×</td>
<td>0.21</td>
<td>0.22</td>
</tr>
<tr>
<td>$n$</td>
<td>43</td>
<td>43</td>
<td>45</td>
<td>×</td>
<td>43</td>
<td>43</td>
<td>45</td>
<td>45</td>
<td>47</td>
<td>47</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>×</td>
<td>45</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 2: Matching probability $p$ for the different bytes of $K_{10}$ and $n$ for $P > 0.99$

Fig. 6: Attack path of the SCA

5.3 Example of Fault Injection Attack (FIA) a on AES

This example is a classic differential fault analysis [4]. The faults are obtained for the same list of 5000 plaintexts as in the previous attack SCA in section 5.2.

- The target is a byte of the key at round 10, $K = K_{10}$. There are only 256 possible values.
- The stimuli is the ciphertext $O_S = C$.
- The reaction is the faulty ciphertext $O_R = C^*$. 
- \( R = g_2 \circ f \circ g_1 \) with: \( f = \) single-bit fault injection process,
  \( g_1 = SB^{-1} \circ SR^{-1} \circ \oplus \) and \( g_2 = \oplus \circ SR \circ SB \).
  The attack path is illustrated in Fig. 7
- \( R_m = g_2 \circ m \circ g_1 \) with 8 possible models, \( \oplus 2^j \) with \( j \in [0, 7] \), considered together.
- The distinguisher is a sieve [4] or a counter [29].
- How many pairs \((C, P_{\oplus 2^j, K_m})\) such that \( i \) is randomly chosen are needed to recover a byte of the \( K_{10} \)? In this case \( \theta = 2.24 \) was claimed in [18]. \( \theta \) is rounded up to 3.
- For the different bytes of the round key \( K_{10} \), the contingency table is generated to compute the matching probability. The \( O_R \) are the faulty ciphertexts observed. The \( P_{m,K} \) are the faulty ciphertexts computed with the models \( \oplus 2^i \). In this precise case the matching probability directly corresponds to the uncertainty of the generated faults when a sufficient number of faults were injected. This value does not depends on the model \( \oplus v \) nor on the key value. As an example, a fault generator which injects two different fault values \((\oplus v_1 \text{ or } \oplus v_2)\) with equal probability would create a contingency table with matching probability \( p = 0.5 \) regardless of the model \( \oplus v \) used. This is why the Table 3 shows the matching probability for only one of the eight models \( (\oplus 2^0) \), the other tables are identical up to a permutation of the columns. Finally, the results are displayed in the Table 3.

<table>
<thead>
<tr>
<th>byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>0.68</td>
<td>0.27</td>
<td>0.35</td>
<td>0.35</td>
<td>0.50</td>
<td>1.00</td>
<td>0.65</td>
<td>0.38</td>
<td>0.69</td>
<td>0.32</td>
<td>0.43</td>
<td>0.68</td>
<td>0.15</td>
<td>0.60</td>
<td>0.74</td>
<td>0.65</td>
</tr>
<tr>
<td>n</td>
<td>6</td>
<td>28</td>
<td>21</td>
<td>21</td>
<td>14</td>
<td>3</td>
<td>10</td>
<td>19</td>
<td>10</td>
<td>23</td>
<td>18</td>
<td>6</td>
<td>54</td>
<td>12</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3: Matching probability for the different bytes of \( K_{10} \) and \( n \) for \( P > 0.99 \)
5.4 Example of a hybrid attack on AES: the Fault Sentivity Analysis (FSA) [5]

A Fault Sensitivity Analysis was performed on the FPGA with the same set-up as the previous fault attack. But when creating a clock glitch, the difference of a normal clock period with the glitchy clock period is a discrete multiple (value named $u$) of the fixed delay 25ps. The value $u$ was stored as the stress of the injection process.
– The target is a byte of the key at round 10, $K = K_{10}$. There are only 256 possible values.
– The stimuli is the faulty ciphertext $O_S = C$.
– The reaction is the sensitivity to the fault $O_R = u$.
– $R = f \circ g$ with $f = \text{Stress}$ which triggers the fault and $g = \text{SB}^{-1} \circ \text{SR}^{-1} \circ \oplus$

  The attack path is illustrated in Fig. 8

– $R_m = m \circ g$ with $m = \text{HW}$.
– The distinguisher is a correlation.
– The question to evaluate $\theta$ is how many pairs $(C, P_{HW,K_{10}})$ are needed in average to recover a byte of $K_{10}$. In this case we have $\theta = 4.0$. $\theta$ is rounded down to 4.
– For the different bytes of the round key $K_{10}$, the contingency table is generated to compute the matching probability. The $O_R$ are the values $u$ and the $P_{m,K}$ are Hamming weights. The results are displayed in the Table 4.

<table>
<thead>
<tr>
<th>byte</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.21</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.20</td>
<td>0.22</td>
<td>0.21</td>
<td>0.20</td>
<td>0.20</td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>46</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>43</td>
<td>46</td>
<td>48</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Matching probability for the different bytes of $K_{10}$ and $n$ for $\mathcal{P} > 0.99$

### 5.5 Comparison of the three attacks

The comparison of the number of experiments $n$ required to have $\mathcal{P} > 0.99$ shows that the presented FIA (section 5.3) is more efficient. Indeed, in average it requires less experiments than the two other attacks. Additionally, for a same attack some bytes of the key are more difficult to extract than others. The difficulty to extract a key byte varies in the different attacks. For example bytes 3 and 13 cannot be extracted with the SCA (section 5.2), but they are easy to extract with the FIA (section 5.3). Moreover, the byte 12 has the biggest $n$ for the FIA (section 5.3), but in the two other attacks it has the smallest. For each attack, the most difficult bytes to retrieve are different.

### 5.6 Reverse Engineering examples

A classic SCARE (Side Channel Analysis for Reverse Engineering) attack as in [30,31] is chosen to illustrate that our formalism is still valid for reverse engineering attacks. An AES-like cryptosystem with customized s-boxes is chosen, i.e. s-boxes are the target. One has to remark that the cipher key is often considered known in a reverse engineering attack. We did not test this attack on our FPGA.
– The target is an s-box \(i.e.\) a boolean function \(b_{8\rightarrow8}\), \(SB(x) = y\). A divide and conquer strategy is achieved by splitting the target into 8 boolean functions \(b_{8\rightarrow1}\).

\[ K = b_{8\rightarrow1} \] There are only 256 possible values.

– The stimuli is the plaintext \(O_S = T\).

– The reaction is the power consumption or the radiation of an electromagnetic field, \(O_R = EM\) or \(O_R = Pow\).

\[ R = f \circ g \] with \(f = EM\) radiation of electromagnetic field or \(f = Pow\) power consumption and \(g = \oplus k_0\).

\[ R_m = m \circ g \] with \(m = HW\) Hamming weight, or \(m = HD\) Hamming distance. The two models are considered separately.

– The distinguisher is a correlation.

### 6 Conclusion

A new technique to compare physical attacks has been presented. This method is able to compare all physical attacks between them, including a side channel attack with respect to a fault injection attack. Additionally using physical attacks in order to find the cipher key or in order to reverse engineer an algorithm do not change the formalism. The main idea is to compare attacks at each step of the attack (campaign, predictions, confrontation) and not only at the end. The improvement comes from the fact that the attack is decomposed in three steps, where the first two are analysed independently. In a first time, only the models are studied with \(\theta\) the average number of queries to the oracle \(\Theta_m\). Then in a second time, only the predictions and the reactions are confronted, without taking into account the distinguishers, with a matching probability \(p\). A smaller \(\theta\) means a better adequacy between the attack path and the model. A bigger \(p\) means a better adequacy between the physical function and the model. Finally a new oracle \(\Theta_p\) which combines both is introduced by creating an oracle with a probability of failure not equal to zero. Two attacks can be compared with the probability of success for \(n\) texts or with the number \(n\) of experiments required to have a probability of success bigger than a chosen value. So if an attack fails, it becomes possible to know if the model is wrong or if the attack path is bad.

It is a new approach which do not confront distinguishers. Instead the comparison is performed before this final step allowing to measure the effectiveness of the other steps. Moreover in a future work we would like to extract from the contingency table some hindsight on what distinguisher is the most likely to work in order to extract a key.

As a perspective, our 3-step approach suggests that by modifying only one of this step, we create a new physical attack. We suggest instead that the components of an attack should be studied independently.

### References


