Evaluating the security of cryptographic systems against electromagnetic attacks

Ricardo Maças¹, Ricardo Chaves¹ and Gonçalo Tavares¹

Abstract—The security of electronic devices is paramount to our information society. Side-channel attacks enable a potential attacker to stealthily compromise the devices by exploiting their emanations, such as the electromagnetic (EM) radiation, resulting from the operation of the device. Two attack techniques are known to successfully compromise devices using EM leakage: correlation attacks and template attacks. The objective of this article is to compare different attack methodologies empirically with respect to their experimental setup and signal processing techniques. A methodology of analysis is defined for consistent and repeatable results. Finally, evaluations of the effects of different experimental setups are conducted, and signal processing techniques are implemented to improve attack performance. In particular, the correlation filtering technique is proposed, which successfully improves the performance of regular correlation attacks in randomized trials for different number of traces.

Index Terms—side-channel attacks, electromagnetic attacks, correlation attacks, template attacks, smartcard

I. INTRODUCTION

The information society is ever more pervasive in our lives: it permeates into all aspects of society, replacing older technology with connected, intelligent devices. More and more devices are made smart, especially the ever more popular Internet of Things (IoT) movement, consisting of existing devices such as door locks that are made intelligent with networking functionality. Some of these devices have important security requirements, and the only good defense against potential adversaries is the study of techniques that can compromise them.

Even if such devices employ secure algorithms, their physical realization is a cryptographic device, implemented with electronics and, naturally, not an isolated system. It needs power to operate, must accept some input commands and return the results using some digital interface. Inevitably, it will have losses such as electromagnetic emanations or can be subjected to faulty behavior when subjected to tampering.

Attacks that compromise the physical security of a cryptographic device by exploiting their leakages are denominated side-channel, due to how they exploit information from channels other than the input and output of the device.

Definition 1: A side-channel attack is an attack that exploits the data-dependent behavior of a cryptographic device to recover the device’s key.

The data dependency is the key phenomenon that allows the enemy to exploit the device. If the emanations depend on the data being processed by the device, then there is information that can be gathered by observation of the signals. In that case, that emanation is called leakage.

II. BACKGROUND

Electronic systems developed in the 21st century use complementary metal-oxide semiconductor transistor cells (CMOS), desirable for producing devices with low idle power consumption. CMOS cells implement a logic gate by having a complementary pull-up and pull-down network that alternate in insulating the path between the power source and the ground, only briefly consuming power during transitions when momentarily both networks conduct. In that moment, a small current is produced and power is consumed. As such, the commutations of the transistors are the main source of power consumption. These can be measured by simply placing a small $1\,\Omega$ resistor in series with the power supply, which varies its voltage drop according to the current flowing from the power supply. As the power consumption varies, so does the voltage measured between the terminals of the resistor.

Naturally, this implies that the power consumption is data-dependent: the transitions triggered in the CMOS cells are related to the change in bits between the original value and the new value. The change in bits between two values is referred to as the Hamming distance between two values, $HD(v_0,v_1)$, where $v_0$ is the initial value and $v_1$ is the new value. However, an attacker prefers the power consumption to depend on the data, not on the difference to a potentially unknown value. In practice, this value may be known, for example, if the bus lines are reset to either ‘0’ or ‘1’ before loading data (called precharging in the latter case). If the bits are all ‘0’, transitions in the bus will only occur for the ‘1’ bits in the new value. The number of bits that are ‘1’ in a binary value is called its Hamming Weight, $HW(v)$. In practice, even if the previous value in the bus is not known nor fixed, there is always some information that can be obtained from a Hamming weight model for power consumption, with enough statistical processing.

The Hamming Weights and Hamming Distance models are two of the most common methods used to model the data-dependent power consumption in side-channel attacks. In general, taking into account some model $h$, the power consumption $P$ can be described as a sum of the contributions of all the gates $g$ in the circuit processing values at instant $t$, affected by some normally-distributed additive noise $N$ [1]:

$$P(t) = \sum_{g} h(g, t) + N(t).$$ (1)
On the other hand, the currents that are generated by the commutation of transistors are also responsible for leaking electromagnetic radiation. While the commutations of the transistors do not generate a static field, for the small electromagnetic field generated near the circuit, called the near-field, the approximation is valid for the observed field amplitude.

There is a well established theory on the relationship between static currents and the magnetic field they induce, a phenomenon first observed by Hans Ørsted. In fact, it can be said that each oriented infinitesimal piece of wire (denoted as $dI$), as it carries some current (denoted $I$) produces a contribution to the magnetic field (denoted $dB$, since it is a differential, i.e., an infinitesimal change in the field) at some point $p$. This contribution depends on the distance and orientation to point of observation, $p$, which can be represented as a vector $\vec{r}$ from the piece of wire to that point. This vector has a norm, which is the distance to the point (denoted $r$) and the direction from the wire to that point (denoted $\hat{r}$, as in, the unit vector pointing in the direction of the vector $\vec{r}$). The $\mu$ constant reflects the fact that the magnetic field depends on the medium where it is emanated, and it is called the magnetic permeability. The law that established this relationship is the Biot-Savard law [2]:

$$dB = \frac{\mu I d\hat{r} \times \hat{r}}{4\pi r^2}$$ \hspace{1cm} (2)

In other words, the magnetic field measured depends primarily on the distance to the source, the intensity of the current and its direction. Recall that, despite the approximations made, this is still a time-varying magnetic field which can be captured using a probe, consisting of a small coil, kept as close as possible to the radiation source in order to minimize the distance. As the probe is subjected to the presence of varying magnetic radiation, a voltage is generated at the terminals of the coil.

None of the previously introduced models takes into account the direction of the current, which is captured by the electromagnetic signals. The Switching Distance model, $SD$, is otherwise identical to the Hamming Distance $HD$, but takes into account the direction of the bit flips. A transition of $0 \to 1$ is considered a positive transition, and increments the Switching Distance by one. On the other hand, a transition of $1 \to 0$ is considered a negative transition and it decrements the Switching Distance by one.

In summary, the electromagnetic and power emanations are related phenomena associated to the same physical process. The same models — Hamming Distance, Switching Distance and Hamming Weights — can be used to estimate the observed signal from the data computed by the circuit.

III. STATE OF ART

There are two main types of side-channel attacks that can be carried out in the power and electromagnetic channels: simple power attacks (SPA) and differential power attacks (DPA).

SPA is simple due to the fact it forsakes processing and uses a small number of traces, perhaps only one, typically analyzed over the time domain by visual inspection. Kocher et al. [3] define it to be "a technique that involves directly interpreting power consumption measurements collected during cryptographic operations".

DPA is differential due to how it compares differences between measured traces and a hypothesis computed with a model of power consumption. It uses statistical analysis so it can reveal a key under considerable noise and without specific knowledge about the cryptographic device, analyzing over a large number of traces but on a small set of time samples, perhaps one at a time. For this it requires a model either informed by the theory of power consumption in cryptographic devices — such as the Hamming Weights model previously discussed — or built statistically with the cryptographic device in question — a template.

While the visual inspection of traces approach used in SPA is useful to locate features to help guide an attack, this article will focus on mainly on DPA techniques. In simple terms, a differential attack starts by capturing and storing the power consumption of a physical device encrypting a given set of known plaintexts — this is called a trace. The remaining steps can be carried away far from the device.

The attack proceeds by taking a working hypothesis over a small part of the key — for example, “is the value of this partial key the number 129?” — and, using the same plaintexts as the physical device, it computes the intermediate calculations of the cipher state. From the intermediate values, using a model of the power consumption of that device, the expected power consumption for those cipher operations is computed.

Then, the differential attack checks if the expected power consumption agrees with the observed signals from the physical device — for example, when the values calculated cause the expected power consumption to increase, does the actual observed consumption also increase? If the differences in the expected power follow the actual power consumption, one has good evidence that the working hypothesis is correct.

Using a statistical test, it is possible to quantify this agreement so that the partial key chosen is the one that displays the best agreement of all tested. This is called the distinguisher, because it is the metric that allows the attack to distinguish the correct secret from the remaining (rejected) hypotheses.

Briefly, the process can generally be described by the following five steps [4], depicted in Figure 1:
1. Choosing an intermediate result of the algorithm and determining how to calculate it from known data, such as the message or the ciphertext;
2. Capturing the traces and corresponding known data;
3. Making hypotheses on the partial key, and calculating the corresponding intermediate values of the traces using the known data;
4. Modeling power consumption based on the hypothetical intermediate values;
5. Comparing the hypothetical power consumption with actual power consumption using a statistical test.

All DPA attacks share these steps but vary on the number of bits attacked, the model used and the statistical test chosen to compare the model with the traces. As per Mangard et al.’s [4] nomenclature, DPA can be classified by their *distinguisher*, even if they are often introduced by their authors as entirely new attack classes. The main DPA techniques are:

- Correlation attacks;
- Template attacks.

### A. Correlation attacks

This attack model was first introduced by Eric Brier et al. [5]. It uses the aforementioned framework for differential power attacks, where the *distinguisher* is the estimator of the Pearson correlation coefficient, which will indicate a linear dependency between the hypotheses for each trace $h_i$ and the power consumption observed at some given time instant $t_0$:

$$r(t_0) = \frac{\sum_{i=1}^{M} (h_i - \bar{h}) \cdot (I_i[t_0] - \bar{I}[t_0])}{\sqrt{\sum_{i=1}^{M} (h_i - \bar{h})^2 \cdot (I_i[t_0] - \bar{I}[t_0])^2}}$$

(3)

where the means are indicated by a bar.

An example 8-bit AES attack to the first round is thus given. AES is the de facto standard cipher, commonly used to encrypt data securely. The AES algorithm starts by adding the original key and then performing the SubBytes operation, which corresponds to doing substitutions to every byte in the state. Focusing only on a single byte of the input plaintext $P_i$, the intermediate byte $I_i$ is calculated by a simple function $I_i = SboxAES(P_i \oplus K_s)$, where $K_s$ is the corresponding key byte. The $SboxAES(x)$ transformation is part of SubBytes in the AES algorithm and is publicly specified.

The device captures the trace $T_i[0...K]$ and some information to help calculate the intermediate value for that trace. In this example, for the first round of AES, the input of the cipher is needed — the plaintext $P_i$. As per the framework, $m$ pairs $(T_i, P_i)$ are stored, $i \in [1...m]$.

An hypothesis on the key byte $K_s$ is made. Based on this key, the intermediate values $I_i$ corresponding to each trace can be computed, and calculating their Hamming weight gives a (relative) estimate of power consumption $h_i$. The values of the hypotheses do not need to accurately describe the power consumption because as long as they are linearly related to the power, the Pearson coefficient will capture their correlation with the traces.

Finally, in the time sample $t_0$ where the intermediate value is calculated in the circuit $t_0 \in [0...K]$, the estimator of the Pearson correlation is calculated between the set of traces at that instant $T_i[t_0]$ and the power model hypotheses $h_i$, $i \in [1...m]$. The process is repeated for each key hypothesis, and the key that yielded the best (highest) absolute correlation is the key byte that the attacker is looking for.

Depending on the device architecture, the power consumption may correlate inversely with the power model. To maximize the information extracted from the attack, the absolute value of the correlation is used instead, which is highest whether the model is directly or inversely correlated.

Should this time instant not be known precisely, the correlation can be calculated for several time samples $t_n$ across the traces and the best (highest) absolute correlation $|r|$ is recorded out of all the time samples considered. In a sense, a *correlation trace $r[t]$* is created which indicates the correlation over time for a given key. As such, the highest absolute correlation out of all the correlation values will be observed in the correct correlation trace for the correct key, at the correct instant where the intermediate value considered is being processed and leaks the most.

### B. Template attacks

The attacker may be interested in compromising some device, but may have limited access to it, only managing to capture a few traces of its execution. If the attacker is able to obtain an identical replica of that device, how could the attacker use it to gain additional information to help their attack? One can leverage the similarity between the two devices in order to build some model of how the device leaks information on its calculations. This model can then be used to compromise the intended device with a small amount of captured traces.

Template attacks are an important class of differential attacks which take advantage of an empirical model — instead of the theoretical models — to reduce the number of traces needed from the device under attack. The attack consists in two phases, as first proposed by Paul Fahn and Peter Pearson [6] in 1999:

1) **Profiling or offline phase**

   The phase of the attack where the power consumption of each operation is characterized using the replica. For example, in AES, there are 256 values possible values that must be modelled for every intermediate byte.

2) **Attack, key extraction or online phase**

   The phase of the attack where the power consumption of the attacked device is compared to the models built in the previous phase.

   The model is obtained by first selecting an interesting time instant, or point of interest, where the value of the calculation being attacked leaks the most. In that time instant, one takes the average value $m$ as an estimator of the distribution mean $\mu$ and the sample variance $s^2$ as an estimator of the variance $\sigma^2$ for the power consumption. The process of building an univariate template will result in $(m, s^2)$ pairs, one for each of the possible intermediate values. In a sense, the mean represents the estimation of the data-dependent part of the...
leakage – the constant term in the leakage model –, while the variance accounts for the noise [7].

Template attacks are typically not understood to only be the inclusion of empirical models in an attack, but also a distinguisher adapted to extract information from such models. This concept was first introduced by Suresh Chari et al. [8] in 2002 based on Signal Estimation theory. Recall the power model presented in Equation 1, with a normally-distributed noise term. Using the mean and variance estimated in the profiling phase, it is possible to model every single value \( i \) as a normal probability density function with those parameters, \( p(\ell|i) \).

When initiating the attack, all the possible intermediate values have the same probability of being correct, described by an uniform probability density \( p(i) \), due to the attacker’s ignorance a priori. To find the correct intermediate values, evidence will need to be collected in the form of traces. Which of the hypotheses are better supported by the measured power consumption leakage \( \ell_n \)?

The likelihood of the measured leakage is, taking the template of the hypothetical intermediate value \( i \), the density of the distribution for the measured power leakage — \( p(\ell = \ell_n|i) \). Taking the previous \( p(i) \) probability for the intermediate value, one can update that probability by multiplying the newly calculated likelihood from the previous step, yielding \( p(\ell|i)p(i) \).

By weighting the likelihood \( p(\ell|i)p(i) \) by the likelihood of the leakage, \( p(\ell) \), the Bayes theorem allows one to derive a new probability density of the intermediate value, but taking into account the observed leakage values, and how likely the leakage is, that is to say, \( p(i|\ell) \), which is the probability of the intermediate value \( i \) having observed leakage \( \ell \).

In summary, to model how some observed leakage \( \ell \) could support the evidence of the intermediate value \( i \), consider the Bayesian posteriori probability function

\[
p(i|\ell) = \frac{p(\ell|i)p(i)}{p(\ell)},
\]

where \( p(\ell) \) is the probability distribution of the observed power leakage, calculated by decomposing it as a sum of the probability functions of the leakage for each of the possible intermediate values \( i_n \). The weighting terms cancel since, as aforementioned, \( p(i_n) = p(i) \forall n \). The following expression is then obtained:

\[
p(i|\ell) = \frac{p(\ell|i)p(i)}{\sum_n p(\ell|i_n)p(i_n)} = \frac{p(\ell|i)}{\sum_n p(\ell|i_n)}. \tag{5}
\]

Notice however this is the likelihood of one intermediate value \( i \) that was hypothesized for a partial key \( k \). In fact, there are several candidate partial keys \( k \), and each key implies \( N \) intermediate values for each of the traces, indexed as \( i_{k,n}, n \in 1 \ldots N \). The attacker would like to model the probability of the partial key, which results from the joint probability density of all the intermediate values hypotheses calculated for that key. Since each trace is an independent event, the likelihood of the key is determined by the product of the \( p(i_{k,n}|\ell_n) \). To apply Maximum Likelihood estimation, it is sufficient to compare which of the partial keys leads to the highest likelihood, as per Equation 6:

\[
\text{arg max } \mathcal{L}(k; i_1\ldots i_n) = \text{arg max } \prod_k p(i_{k,n}|\ell_n). \tag{6}
\]

Even though the distinguisher used is different from those used in the Background section, the same DPA principles still apply for all the remaining steps of the attack.

### IV. METHODOLOGY

The definition of the experimental setup begins with the metrics used to evaluate the empirical results. After, the framework to carry out the attacks is presented. Finally, the details of the experimental setup are presented.

#### A. Success metrics

It is of the utmost importance to define what success means. The uncertainty associated with the signal variability and the statistical tools used means that two attacks using signals captured in the identical conditions do not necessarily have the same results, so it becomes important to define what it means for an attack to be successful in a systematic and reproducible manner.

Since attacks may vary, each attack combination will be tried at least 50 times and the results will then be computed using a set of well-defined metrics. The metrics are presented with their average \( m \) and an error term \( e \), called the standard error, which characterizes the uncertainty of the average value for one standard deviation of the sampling distribution (approximately 68%). The standard error for the average is calculated using the standard deviation of the results \( s \) as follows:

\[
e = \frac{s}{\sqrt{N}} \tag{7}
\]

where \( N \) is the number of trials which were used to compute the sample mean. For example, there are 16 partial keys in an AES attack, so, when 50 complete trials are run, \( N = 16 \times 50 = 800 \) partial trials are executed.

Differential attacks return a distinguisher score for every candidate partial key tested. If these scores are sorted in increasing order, their position in the candidate list indicates their relative rank amidst themselves. As such, every candidate has a score and position pair \((d_i, p_i)\).

One way of determining the performance of the attack is by testing the attack methodology under a controlled experimental setting, knowing in advance which of the possible values for the subkey is the correct one. The correct value pair will be denoted by a star, as \((d_1, p_1)\). An immediate question follows: where did the attack place the correct value of the subkey?

1) Success rate (SR): The most obvious course of action is to check whether the attack determined the correct value as the best candidate, that is, if \( p_\bullet \equiv 1 \). In that case, the attack to that partial key was successful. However, the success of a single attack does not adequately represent its performance and it is necessary to repeat it several times, \( N \). The success rate can then be calculated as the ratio of attacks, ran under the same conditions, that yielded \( p_\bullet \equiv 1 \):

\[
SR = \hat{P}(p_\bullet = 1) = \frac{\# \text{Attacks}_{p_\bullet = 1}}{N}. \tag{8}
\]
The success rate gives an estimate of the probability $\hat{P}$ of having the correct value of the subkey as the best candidate, for an attack ran under the same conditions as the ones used to estimate it. [9]

2) Average correct position and Guessing Entropy (GE): The correct value position can be interpreted in a different way: if the correct value is the $n$th one, then the attacker would need to test all keys up to $n$ to find the correct value.

After collecting $N$ attacks ran under the same conditions, one may estimate what is the mean position of the correct partial key in the candidate list. Consider the notation $E(\cdot)$ as representing the expected value, also known as expectation. The expected correct position is then defined as the product of each position $i$, times the probability of the correct subkey occupying such position, or mathematically,

$$E(p^*) = \sum_i i \cdot P(p^* = i).$$

Then, the estimator of the expected correct position can be mathematically expressed as the average correct position $\tilde{E}(p^*)$, computed as

$$\tilde{E}(p^*) = \frac{\sum_j^N p^*(j)}{N}.$$  (10)

where $j$ represents each attack out of $N$ total attacks.

This metric was named by Standaert et al. [9] as the guessing entropy. The nomenclature guessing entropy is related to the fact that the number of positions that have to be tested to guess the correct key is, in a sense, a measure of uncertainty or missing information.

Since entropy is more often measured in bits, other authors [10] [11] usually define it as being the logarithm base 2 of the previous metric. As this is usual practice, the Equation 10 will be called the average correct position, while the guessing entropy will be thus defined as

$$GE = \log_2(\tilde{E}(p^*)) \text{ bits.}$$  (11)

3) Correct normalized distance: Another way of evaluating success is based on the confidence of the result given by the statistical test. In order to evaluate success, one must ask: to what extent does the distinguisher distinguish the correct candidate from the rest?

To translate this into a mathematical concept, a metric is defined that encodes the relative difference between the correct candidate $d^*$ and the second candidate $D_2$. The metric was given the name normalized distance.

$$w = \frac{d^* - D_2}{D_2} = \frac{D_1}{D_2} - 1.$$  (12)

For example, a normalized distance of 0.3 means that the correct candidate has a distinguisher score 30% higher than the second best candidate. The degree of separation is defined to be completely independent of the kind of distinguisher used: the normalized distance refers to a degree of separation whether it is applied to correlation, likelihood or a probability. Recall that different statistical tests have different sensibilities, so, for different types of distinguisher, while the normalized distance means the same, it must be interpreted differently.

A similar metric, the distance between the correct and best wrong candidate weighted by the standard deviation of the candidates’ distinguishers, is presented in Whitnall and Oswald. [12]. The authors named it Nearest-rival distinguishing power. Such a measure is effectively independent of the type of distinguisher used, which addresses the aforementioned critic on non-normalized distances.

4) Success rate for a complete key: Whichever strategy used, the best guess for the complete key is trivially the combination of the best guesses for each of the partial keys. The success rate for a complete key ($SR^{CK}$) is redefined as the ratio of attacks (out of $N$ total attacks) that simultaneously reported all the correct $i=1,...,K$ subkeys in the first position:

$$SR^{CK} = \frac{\# \text{Attacks}_{p^*_i \leq 1, \forall i = 1,...,K}}{N}.$$  (13)

This is equivalent to the joint success rate of every partial key $SR^i$. That is to say, the probability that the best guess of the complete key is the the correct key, $SR^{CK}$, is simply the product of the probabilities that our best guesses for the $K$ partial keys are correct, or mathematically,

$$SR^{CK} = \prod_i SR^i, \forall i = 1,...,K.$$  (14)

B. Experimental framework

Three main components comprise the framework for running a side-channel attack: the device under attack, which executes the targeted cryptographic algorithm with a hidden key, the attacker device, responsible for executing the computations needed to reveal the correct key and coordinating the attack, and the signal acquisition device that samples the electrical signals in its input and generates a digital signal that can be processed at the attack device. The entire process flow is represented in Figure 2.

First of all, the signal acquisition device was chosen, a PC oscilloscope named the PicoScope 6404D. This oscilloscope is rated with 500MHz (-3dB) bandwidth with a maximum sampling frequency of 5 GHz and 8 bits per sample. The oscilloscope samples the signals at the input ports and outputs...
the data stream via an universal serial bus (USB) interface to the computer.

There are two possible devices under attack (DUA). There is a reference board, the SAKURA-G R1 [13]. This reference board is an FPGA (Field-Programmable Gate Array) circuit which includes a chip that executes the cryptographic routines, and components that facilitate performing the attacks and allow for the execution of the experimental work in repeatable and well-standardized conditions. The board also includes internally amplified and non-amplified power consumption output ports that allow for simple execution of power attacks.

The second device under attack is the smartcard. The smartcard is a programmable portable chip that can execute cryptographic routines. As before, an expansion board SAKURA-W is used to facilitate the execution of the attacks under consistent conditions. The expansion board includes a power consumption output port, connected to the smartcard, which can be used to perform the smartcard attacks.

Both devices under attack were programmed with the AES cipher and a triggering mechanism, which allowed the oscilloscope to capture time-aligned traces easily. In power and electromagnetic attacks, the source of power is very important. Any noise introduced by the power source is inevitably included in the power and electromagnetic traces. The devices were powered by a laboratory bench power supply for a stable and isolated power source.

Finally, the attacker device is a personal computer (PC). The capture process, the attacks and signal processing techniques will be all be implemented by the computer. Software was developed, based on the previous work by João Lopes [14]. To gather the traces and interface with the oscilloscope and the device under attack, the capture script was developed in Python. The attack scripts, taking as input the traces captured in the previous script, were developed in MATLAB to take advantage of its powerful matrix manipulation capabilities, in order to cope with the demanding computation tasks over large amounts of data, necessary to carry out the attacks.

V. POWER RESULTS

First, power attacks were carried out. This choice is motivated by the fact that existing previous work [14] and infrastructure at the research facility. This will provide the practical know-how needed in side-channel attacks to validate the techniques which will then be used in electromagnetic attacks. Attacks were run targeting both the FPGA and the Smartcard, to evaluate the suitability of each device for EM attacks.

A. FPGA Results

The first test was conducted in order to evaluate the performance of three different experimental setups: attacks using the power channel from the internal 20 dB amplification provided by SAKURA-G (iAMP), attacks using the power channel from an external amplifier set at 20 dB (AMP), and finally a setup using both in series (iAMP+AMP). This comparison will indicate which is the best baseline configuration in order to test the remaining attacks.

1) Trace filtering: Trace filtering is the technique characterized by the application of a low-pass filter to the traces in order to filter sudden signal variations mostly caused by noise. An attacker may have a clone of the device she or he intends to attack, and may use it to identify which low-pass filter provides optimal results.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>( SR )</th>
<th>( SR^{C,K} )</th>
<th>( w )</th>
<th>( E(f^*) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>50 000 traces</td>
<td>400 retrials</td>
<td>0.703 ± 0.006</td>
<td>0.000 ± 0.000</td>
<td>0.138 ± 0.002</td>
</tr>
<tr>
<td>50 000 traces</td>
<td>50 retrials</td>
<td>0.75 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.15 ± 0.01</td>
</tr>
<tr>
<td>50 000 traces</td>
<td>50 retrials</td>
<td>0.70 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.12 ± 0.01</td>
</tr>
</tbody>
</table>

In conclusion, the externally amplified setup provides the clear advantage. The dual-amplified setup, with a significantly increased signal amplitude, did not improve the results when compared to either the single internal or external amplifier, given a normalized distance and correct partial key position which are not statistically different from the attacks run against the internally amplified signal. As a result, the externally amplified setup will from now on be used as the reference setup.

Taking into account the setup results and the side-channel theory, two signal processing techniques based on filtering were considered in order to further improve the results of the best setup: trace filtering and correlation filtering. Knowing most of the leakage energy is located in the lower frequencies, 50 low-pass filters were designed using MATLAB’s fir1( ) command, with a \(-6\) dB cut off frequency varying between 0.001 and 0.050 in frequency normalized to half the sampling rate.

Fig. 3. Example of two consecutive filters generated. The filters generated have cutoffs shifted 0.001 in the normalized frequency, at \(-6\) dB.
Using correlation attacks, a frequency sweep allowed the identification of the cut-off frequencies that yielded the best results for the attacks using the low-pass filters. The sweep is displayed in Figure 4, and two cut-off frequencies interest resulted in the lowest average correct partial key position.

![Estimation of optimal low-pass cutoff frequency for trace filtering](image)

Fig. 4. Results of correlation attacks run against traces processed using filters of different cutoff frequencies (50 trials with the Hamming weights model, 1.25 to 62.5MHz).

The two frequencies selected were 3.75 MHz and 13.75 MHz. The performance of the trace filtering for these two cut-off frequencies in correlation attacks is displayed in Table II.

### Table II

<table>
<thead>
<tr>
<th>Experiment</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50 000 traces</td>
<td>SR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>regular</td>
<td>0.75 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.15 ± 0.01</td>
</tr>
<tr>
<td>AMP</td>
<td>0.75 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.23 ± 0.01</td>
</tr>
<tr>
<td>50 000 traces correlation filtered</td>
<td>0.77 ± 0.01</td>
<td>0.00 ± 0.00</td>
<td>0.23 ± 0.01</td>
</tr>
<tr>
<td>13.75 MHz</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Trace filtering provides a measurable improvement in performance when compared to the reference attack, with improvements in the average correct key position and the normalized distance. However, in the next experiment, a similar technique is proposed which further improves the results.

2) **Correlation filtering**: Correlation filtering is a proposed technique for correlation attacks that takes the correlation over time of every value hypothesis and filters sudden variations, to remove sporadic correlation peaks from competing hypotheses and to ensure that the correlation smoothly peaks at the time instant when the operation is performed. For the typical criteria of choosing the correlation maxima, the noise fluctuations may easily be preferred instead of a correct key. Depending on the device under attack, the attacker might want to choose a low-pass filter that preserves the leakage while improving the results.

As in the trace filtering case, a study of the optimal filter to use was conducted by testing the effect of filtering the traces with each of the available filters using correlation attacks under the same conditions. The optimal filter selected, with cut-off frequency of 18.75 MHz, displayed the best average correct partial key position.

Table III compares the results of the correlation filtering using both filters to the reference setup.

### Table III

<table>
<thead>
<tr>
<th>Experiment</th>
<th>50 000 traces</th>
<th>200 000 traces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Attack</td>
<td>0.81 ± 0.01</td>
<td>0.00 ± 0.00</td>
</tr>
<tr>
<td>Correlation Attack</td>
<td>0.75 ± 0.02</td>
<td>0.00 ± 0.00</td>
</tr>
</tbody>
</table>

Using the proposed technique, a statistically significant improvement is achieved when compared to trace filtering and the original reference.

### B. Smartcard results

In the case of the smartcard, the template and correlation power attacks were compared by running attacks under the same conditions and using the same traces.

For the correlation attacks, the Hamming Weights model was used. In the case of the Template Attacks, an empirical model was built using a total of 1 000 traces, or in other words, 3.9 samples per intermediate value, on average. The samples were selected from the points of interest that were obtained when running the correlation attacks.

### Table IV

<table>
<thead>
<tr>
<th>Experiment</th>
<th>20 traces</th>
<th>50 retrials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Attack</td>
<td>0.879 ± 0.008</td>
<td>0.110 ± 0.031</td>
</tr>
<tr>
<td>Correlation Attack</td>
<td>0.54 ± 0.02</td>
<td>0.00 ± 0.00</td>
</tr>
</tbody>
</table>

The template attack shows promising improvements when compared to the correlation attack, improving in all metrics by a statistically significant degree.

### VI. EM Results

Based on the previous acquired knowledge in the execution of power attacks, new experiments were carried out in the context of EM attacks in order to evaluate the success of the attacks and some signal processing techniques.
First, the specific details of the EM experimental setup are detailed. The probe is the device responsible for the conversion of the electromagnetic radiation into an electrical signal. In this case, a near-field attack is proposed, using magnetic field probes. The Beehive Electronics 100B probe will be the preferred choice for the attacks, with a small loop size and an increased spatial resolution.

A probe amplifier is necessary to obtain traces with enough power to be captured by the acquisition device reliably. The Beehive Electronics 150A EMC probe amplifier is used to amplify the signal from the probes by 32 dB (± 1.5 dB), in the frequency range of 100 kHz to 6 GHz, with a noise figure between 5 and 6 dB.

In order to allow for separation of the smartcard from the reader circuit, and to facilitate the placement of the probe, a smartcard extender was developed. A smartcard extender is a custom-built printed circuit board that facilitates the EM attacks, depicted in Figure 5. It includes a cutout that allows the placement of the probe directly over the chip area of the smartcard.

![Smartcard Extender Image](image)

Fig. 5. Picture of the smartcard extender. A smartcard was placed in the extended slot, and the cutout reveals the surface under which the microprocessor resides.

1) Probe positioning: A systematic analysis of the circuit hotspots was carried out, consisting of two steps: selection of signals and verification of leakage using t-test analysis.

The smartcard was set to carry out sequential random encryptions. Meanwhile, the magnetic probe was placed directly on top of the chip area and the surface was scanned while monitoring the signal displayed by the oscilloscope on the probe channel. One expects the leakage signal to be fairly periodic, with some relation to the clock signal, while exhibiting some data-dependent variation in amplitude. Interesting hotspots were identified and the collection of traces was carried out for the fixed vs. random t-test.

The second step, t-test analysis, was carried out to validate the existence of leakage. In a fixed vs. random t-test, both a fixed and a random set must be generated. For the random set, random subkeys and plaintexts were generated.

One spot showed very good leakage, where the different operations of the first round were clearly visible, depicted in Figure 6. Some of the locations chosen displayed a strong signal, but in the verification step were not exhibiting significant leakage. Further analysis concluded that the intense signal observed was due to interference from the communication pins on the smartcard, which radiates considerably. This was verified by extending the duration of the capture and verifying the presence of a square wave noise that most likely was due to data transmission. As such, the attacker must be careful not to assume the correct signal is the strongest, and should always proceed with a careful analysis to verify leakage locations.

2) Model comparison: Using the smartcard extender and the probe in the optimal position, the Hamming Weights and the Switching Distance model were compared in correlation attacks. In particular, for the Switching Distance model, the best results were achieved when taking the distance to the partial key. The comparison of the results is shown in Table V.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>SR</th>
<th>SRCK</th>
<th>w</th>
<th>E(pw) (% impr.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 000 traces</td>
<td>0.57 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.06 ± 0.01</td>
<td>8.36 ± 0.75</td>
</tr>
<tr>
<td>2 000 traces</td>
<td>0.55 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.07 ± 0.01</td>
<td>7.85 ± 0.70 (6.03%)</td>
</tr>
</tbody>
</table>

A modest improvement in performance is achieved with the Switching Distance model. Due to the small scale of this improvement, the reference attack will be taken as the Hamming Weights attack.

3) Correlation filtering: As in the power attacks, a correlation filtering attack is carried out, starting with a frequency sweep to identify the best cut-off frequency. In the case of the EM channel, the best cut-off frequency was determined as 16.25 MHz out of all the existing filters. The results of these attacks are displayed in Table VI.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>SR</th>
<th>SRCK</th>
<th>w</th>
<th>E(pw) (% impr.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 000 traces</td>
<td>0.57 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.06 ± 0.01</td>
<td>8.36 ± 0.75</td>
</tr>
<tr>
<td>0.5000 Hz</td>
<td>0.98 ± 0.02</td>
<td>0.70 ± 0.02</td>
<td>1.00 ± 0.00 (88.0%)</td>
<td></td>
</tr>
</tbody>
</table>

The correlation filtering yielded consistently better results than the reference attack, achieving near-unity success rate for the complete key at a number of traces that, in the reference attack, did not result in any successful attack for a complete key. Independently of the channel used, the proposed correlation filtering technique increases performance for the sampling frequencies considered.

4) Additional amplification: For a final test of correlation attacks, additional amplification was used. This was carried out by capturing the signal using an external amplifier, set at a gain of 18 dB, with the regular use of the probe amplifier. The comparison is laid out in Table VII.

Unlike in the power attack, the additional amplification improved the results of the attack significantly when compared...
Fig. 6. T-test trace (intermediate values fixed at zero versus random) for the EM leakage at the final probe position.

![Smartcard EM t-test: first round AES](image)

**Fig. 7.** EM templates built for one of the intermediate state bytes.

The comparison presented suggests that the performance of the template attacks is worse than the performance of correlation attacks. Template attacks are claimed to be “the strongest form of side channel attack possible in an information theoretic sense” [8], but inevitably depend on the model built with the templates and the points of interest that are used. Attempts were made to improve the results, retraining the templates with 50 samples per intermediate value instead of approximately 30, and changing the points of interest to those obtained from correlation attacks. However, these efforts did not seem to improve the results significantly in small scale testing.

### Conclusion

The security of electronic devices is paramount to our information society. The majority of the devices already implement standardized secure algorithms, but suffer from implementation issues. Side-channel attacks enable a potential attacker to stealthily compromise the devices in question. In particular, electromagnetic side-channel attacks can be carried out using techniques such as the ones described in this paper.

**TABLE VII**

**Comparison of Results for the Setup with Additional Amplification in Electromagnetic Correlation Attacks to the First Round of AES, for 50 Retrials (Sampled from 5500 Traces).**

<table>
<thead>
<tr>
<th>Experiment</th>
<th>SR</th>
<th>$SR_{E^K}$</th>
<th>w</th>
<th>$E(p_*)$ (% impr.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000 traces</td>
<td>0.57 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.06 ± 0.01</td>
<td>8.36 ± 0.75</td>
</tr>
<tr>
<td>2000 traces</td>
<td>0.83 ± 0.01</td>
<td>0.08 ± 0.04</td>
<td>0.15 ± 0.01</td>
<td>1.60 ± 0.09 (80.90%)</td>
</tr>
<tr>
<td>50 retrials</td>
<td>0.57 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.06 ± 0.01</td>
<td>8.36 ± 0.75</td>
</tr>
<tr>
<td>500 samples</td>
<td>0.255 ± 0.011</td>
<td>0.000 ± 0.000</td>
<td>-0.001 ± 0.000</td>
<td>25.268 ± 1.049 (-202.2%)</td>
</tr>
<tr>
<td>10000 samples</td>
<td>0.255 ± 0.011</td>
<td>0.000 ± 0.000</td>
<td>-0.001 ± 0.000</td>
<td>25.268 ± 1.049 (-202.2%)</td>
</tr>
</tbody>
</table>

The comparison presented suggests that the performance of the reference attack. This suggests that the amplification is useful when the signal does not occupy the entire range of the oscilloscope, and as such, adjustable amplifiers should be used to ensure the captured signal occupies the entire bit depth allowed by the signal acquisition device.

5) Template attacks: Template attacks provided improvements for smartcards attacked using the power channel. Taking into account that the power channel and the electromagnetic emanations are related phenomena, a similar improvement is expected.

As before, for the correlation attacks, the same reference attack using the Hamming Weights model was used. In the case of the Template Attacks, an empirical model was built using a total of 7600 traces, with 29.97 samples per intermediate value, on average. The samples were selected from the points of interest that were obtained when running the t-test. The comparison is presented in Table VIII.

**TABLE VIII**

**Comparison of Results for Different Attack Techniques in Electromagnetic Attacks to the First Round of AES.**

<table>
<thead>
<tr>
<th>Experiment</th>
<th>SR</th>
<th>$SR_{E^K}$</th>
<th>w</th>
<th>$E(p_*)$ (% impr.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000 traces</td>
<td>0.57 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.06 ± 0.01</td>
<td>8.36 ± 0.75</td>
</tr>
<tr>
<td>500 samples</td>
<td>0.57 ± 0.02</td>
<td>0.00 ± 0.00</td>
<td>0.06 ± 0.01</td>
<td>8.36 ± 0.75</td>
</tr>
<tr>
<td>10000 samples</td>
<td>0.255 ± 0.011</td>
<td>0.000 ± 0.000</td>
<td>-0.001 ± 0.000</td>
<td>25.268 ± 1.049 (-202.2%)</td>
</tr>
</tbody>
</table>

The security of electronic devices is paramount to our information society. The majority of the devices already implement standardized secure algorithms, but suffer from implementation issues. Side-channel attacks enable a potential attacker to stealthily compromise the devices in question. In particular, electromagnetic side-channel attacks can be carried out using techniques such as the ones described in this paper.
In an adversarial context, such a mechanism is unlikely to be available, and as such, the development of an alignment technique that does not depend on external triggering is proposed. There are several well-known techniques that can enable this, such as cross-correlation, and the development of software to take advantage of such techniques would enable their application under an experimental setting. In recent years, new side-channel techniques have been published inspired by machine learning. There are several techniques in this area that are able to do classification of hypotheses and to learn power models, which promise to bring improvements to the attacks. For future work, the characterization of these new techniques and application of signal processing tools is proposed.

Finally, there are several countermeasures developed to minimize the impact of side-channel attacks in commercial devices. Some of these countermeasures, such as masking, have known attack strategies which can be used to overcome them. The evaluation of these techniques in electromagnetic attacks, and the application of relevant signal processing techniques, would be a valuable contribution.

**REFERENCES**


Ricardo Maçãs is a student majoring in Telecommunications (with a minor in Computers) in the Master in Electrical and Computer Engineering degree at Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal.