Collision Based Attacks in Practice

Ibrahima Diop, Pierre-Yvan Liardet, Yanis Linge, Philippe Maurine

Collision Based Attacks in Practice

Diop Ibrahima
STMicroelectronics and
Ecole Nationale Superieure
des Mines de Saint-Etienne (EMSE)

Pierre-Yvan Liardet
and Yanis Linge
STMicroelectronics

Philippe Maurine
LIRMM, Université Montpellier II and
Commissariat à l'Energie Atomique
et aux energies Alternatives (CEA)

***Abstract*—Chosen-Message Simple Power Analysis, also called Collision Based Attacks (CBA), have been proposed by Fouque, Yen and Homma. These attacks aim at inducing and detecting collisions during modular operations. However, detecting collisions is a challenging task in real environments. Doing it in an automated manner is even more challenging. In this paper, we propose and compare some methods and criteria allowing to automatically (without any visual inspection) detect the occurrence of collisions in leakage traces acquired on modern (and thus noisy) circuits.**

I. Introduction

Modular exponentiation plays a fundamental role in public key cryptosystems, such as RSA [18]. Unfortunately the implementation of most of the modular exponentiation algorithms are vulnerable to Side Channel Attacks (SCA) exploiting the power consumption [14], the electromagnetic radiations [8] or the computation times [13] if the designers does not implement ad-hoc countermeasures.

Over the last few years, several works have highlighted the threat constituted by chosen message attacks which can be viewed as an enhanced Simple Power Analysis (SPA) [7], [24], [23], [9], [10], [20] or as a Differential Power Analysis [14] conducted in an horizontal manner [5], [22], [4], [2]. The related techniques, categorized under the terms 'Comparative Power Analysis' or 'Collision Based Attacks' (CBA), compare two segments (time frames) of two leakage traces and then use the results in order to decide whether the operation processed during these two time frames are the same or not, i.e. to decide whether there is a collision or not.

To be able to interpret the occurrence of a collision, i.e. to reveal some secrets, the occurrence of a collision must be controlled but must also depend on a secret information value (e.g. an exponent bit value). At this aim, the idea is to choose message pairs expected to produce a collision at known steps of private or public key algorithms, i.e. at some known (adjacent or not) leakage patterns of the two related SCA traces.

To apply a CBA, the execution of the target cryptographic algorithm must be regular in time because adversaries must be able to distinguish one leakage pattern from another. Except this constraint which is met on public key algorithms to thwart SPA, CBA is considered applicable to all standard exponentiation algorithms including the binary methods [16], m-ary (window) methods [12], sliding window methods [12], and their variants [12], which are resistant to previously proposed attacks [14].

As a result, CBA has been demonstrated efficient against symmetric cryptographic algorithms in [19], [3], and asymmetric cryptographic algorithms using modular exponentiation in [7], [24], [9] as detailed in section II.

However in real environment, the collision detection is a challenging task because of trace misalignment and reduced Signal-to-Noise Ratio (SNR). This is especially true if one aims at doing it in an automated and relevant manner on modern micro-controllers or smart-cards. In order to meet this challenge, Chen et al. proposed in [1] an automated solution based on clustering while in [17] Perin et al. defined a detection criterion in order to ease the detection of collisions. Within this context, this paper tackles the collision detection problem by combining and comparing some solutions to finally introduce a practical and efficient solution. The latter is based both on leakage trace compression and on a bounded collision detection criterion.

The rest of the paper is organized as follows. In section II, some preliminaries related to the RSA implementations are given as well as a summary of previous works on Collision Based Analysis applied to public key cryptosystems. Section III proposes an analysis of some general and specific techniques that can ease the detection of collisions. Additionally a bounded collision detection criterion is also introduced and justified in this section. Experimental results using the different techniques and the proposed collision detection criterion are given in section IV. It is important to notice that experiments described in section IV were done with one shot leakage traces measured on modern micro-controllers. This constraint was imposed by the jitter of modern circuits we considered but also in view of the frequent use of countermeasures such as shuffling and dummy operation insertion. Finally a conclusion is drawn in last section.

II. Preliminaries, related works

A. RSA and modular exponentiation

1) The RSA cryptosystem: RSA is an algorithm for public-key cryptosystem and was first publicly described by Rivest, Shamir and Adleman in 1978 [18]. The RSA algorithm is designed as follows:
Let n be a product of two large prime integers p and q.
To cipher(resp. to decipher) a message M (resp. a cipher C) with RSA, the algorithm computes $C = M^e \ mod \ n$ (resp. $D = C^d \ mod \ n$) where e is a public exponent (resp. d is

private exponent). The security of the algorithm stems from the hard problem of factoring the modulus n.

2) The Binary exponentiation method: The binary method is an efficient exponentiation algorithm. It performs multiplication and squaring operations according to the bit pattern of the exponent. There exist two variations of the algorithm. The left-to-right binary method (algorithm 1), which starts from the exponent's MSB (the Most Significant Bit) to its LSB (the Last Significant Bit), and the right-to-left binary method (algorithm 2), which operates in the opposite direction. In these algorithms k is the bit length of the secret key.
Because of its higher performance and low resource requirements the left-to-right binary method is the mostly used. We therefore consider only this exponentiation method.

Algorithm 1 LEFT-TO-RIGHT BINARY METHOD

Require: $M, d = (d_{k-1}, ..., d_1, d_0),\ d_i \in \{0,1\}\ \forall\ i,\ n$
Ensure: $C = M^d\ mod\ n$
1: $C = 1$
2: **for** $i = k-1\ \ downto\ \ 0$ **do**
3: $\quad C = C \times C\ \ mod\ \ n$
4: $\quad$ **if** $d_i = 1$ **then**
5: $\quad\quad C = C \times M\ \ mod\ \ n$
6: $\quad$ **end if**
7: **end for**
8: **return** C

Algorithm 2 RIGHT-TO-LEFT BINARY METHOD

Require: $M, d = (d_{k-1}, ..., d_1, d_0),\ d_i \in \{0,1\}\ \forall\ i,\ n$
Ensure: $C = M^d\ mod\ n$
1: $C = 1$
2: $B = M$
3: **for** $i = 0\ \ to\ \ k-1$ **do**
4: $\quad$ **if** $d_i = 1$ **then**
5: $\quad\quad C = C \times B\ \ mod\ \ n$
6: $\quad$ **end if**
7: $\quad B = B \times B\ \ mod\ \ n$
8: **end for**
9: **return** C

B. Collision Based Analysis and public key algorithms

One of the first examples of a CBA applicable to public key cryptographic algorithms is the Doubling Attack (DA). It was introduced by Fouque and Valette in [7]. The DA requires acquiring two leakage traces during the computation of two modular exponentiation with two chosen messages. Typically, an attacker uses:

$$M\ mod\ n \text{ and } M^2\ mod\ n,$$

as input messages. The secret is disclosed by detecting or not a collision between a squaring operation at $(i+1)^{th}$ iteration loop in the leakage trace of M and a squaring operation at the i^{th} iteration loop in that of M^2. Indeed, the collision occurs if and only if the equivalent exponent bit d_i is 0. Figure 1 illustrates the principle of the DA which is applicable to any modular exponentiation based on a left-to-right binary method even if the Device Under Test (DUT) features one of the popular blinding countermeasures [6].

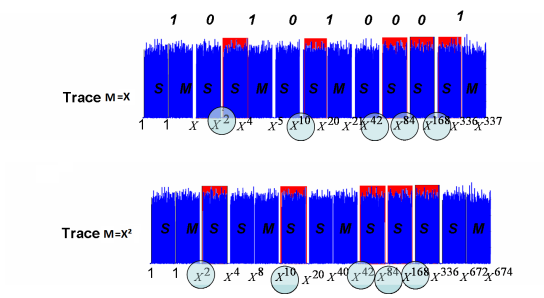


Fig. 1: Illustration of the principle of the Doubling Attack

Yen et al. extended the DA in [24]. They proposed to choose a single input:

$$-1\ mod\ n,$$

as the chosen message to induce a strong differences between the processing of a modular multiplication and of the modular squaring. They also discussed in their paper the possibility of defeating one of the most popular SPA countermeasure: the insertion of dummy multiplications [6].

Yen et al. also proposed in [24] to exploit the two leakage traces associated to the processing of:

$$M\ mod\ n \text{ and } -M = (n - M)\ mod\ n$$

For the rest of this paper we will refers to this attack as Opposite Attack (OA). This proposal is entrenched in the fact that the processing of M and $-M$ are identical if $(d_{k-1}...d_0)_2$ is an even integer after the iteration i. Therefore, d_i can be identified by detecting the occurrence or not of a collision (Figure 2).
In the above, the processing of the chosen messages induces a collision of squaring at the adjacent or at the same time frames in the two related leakage traces and can only be applied to implementations based on the left-to-right binary method.

In contrast, the CBA proposed by Homma et al. in [9], [10], can generate a collision between two leakage traces at two arbitrary time frames by choosing two input messages in a more flexible relationship. The idea is to use two input messages Y and Z satisfying the equation:

$$Y^{\alpha} = Z^{\beta}\ mod\ n,$$

where α and β are constants. The value of α is chosen to provide information on a certain bit, and $\beta \leq \alpha$ is chosen arbitrarily. This attack requires two leakage traces to disclose each bit of the exponent. Contrarily to their first proposal, this attack can be applied to all standard exponentiation algorithms. Some further extensions have recently been proposed by Yen et al. [20] who provide some simulated results of their attacks.

By considering the attacks we have shortly described, one may thus conclude that literature provides several efficient

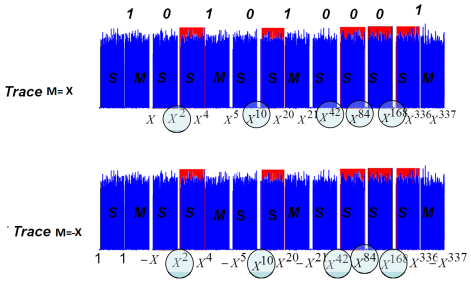


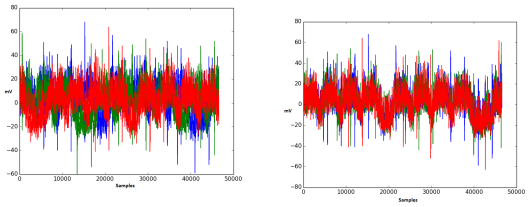
Fig. 2: Illustration of the principle of the Opposite Attack

solutions to induce collisions. However, to the best of our knowledge there are only few works proposing solutions to detect efficiently and automatically collisions; a key problem to enable an efficient practice of CBA on modern smart cards or micro-controllers. Among them, Chen et al. recently proposed in [1] an automated solution based on clustering while in [17] Perin et al. defined a detection criterion to ease the detection of collisions in practice. Within this context, this paper extends these methods in order to automatically detect the occurrence of collision in leakage traces acquired on modern (and thus noisy) circuits.

III. Criteria and Practical techniques for Collision Based Attacks

In real environment, the collision searching is a challenging task because of trace misalignment, reduced Signal-to-Noise Ratio (SNR) or because of the occurrence of outlier events in the observed trace recorded (e.g. the firing of a pump charge). Countermeasures, like the random insertion of dummy operations and shuffling, are just some of the additional and possible reasons [15] explaining the difficulty of detecting collision. The fact whether the recorded leakage traces are aligned or not, essentially depends on the signal used (usually the communication I/O of a smart card) to trigger the oscilloscope. Several techniques were proposed to improve SCA efficiency in presence of misaligned leakage traces. In [11] a high resolution waveform matching method based on Phase-Only Correlation (POC) is proposed. this method is particularly efficient because of its robustness to outliers (Peaks included by the firing of a pomp charger). This explains why we privileged it in this paper. The POC computes the difference of phase between two traces using the cross-phase spectrum of the Discrete Fourier Transform (DFT) of these traces (see [11] for survey).

As an illustration, Figure 3a shows three traces we measured above a modern micro-controller before application of any alignment method. Figure 3b shows the same traces after application of the POC. As shown, even if the leakage traces collected above a modern micro-controller are quite noisy, POC is efficient in suppressing (or in reducing) the misalignment between traces. This technique is therefore considered as an unavoidable pre-processing step in the remainder of the paper; pre-processing systematically applied prior to launch any collision detection technique.



(a) Measured power traces on a product before application of any alignment technique

(b) Measured power traces on a product after application of the POC

Fig. 3: Measured power traces on a product before and after application of alignment technique

A. Collision Detection Techniques

Chen et al. proposed in [1] an automated solution based on clustering to automatically detect the occurrence of a collision. However, their iterative solution requires the definition of a threshold value by the adversary. The definition of this a priori unknown threshold constitutes a limit of their proposal. Indeed, fixing the threshold value to a too high or low value can lead to long processing times or to a poor discrimination capability of collisions.

B. Detection Criteria

More recently Perin et al. defined a collision detection criterion in [17]. This criterion, the $PCDC$, is based on engineering intuition. Let us remind the basis on which is entrenched this criterion. At this end let:

- $T_i = [t_1^i, \cdots, t_w^i]$ be a vector of samples, collected with an oscilloscope, representing the leakage pattern i. This pattern could correspond to a squaring (resp. a multiply), in that case we use to denote it by T_S (resp. T_M),
- $P(T_E, M, d)$ represent the leakage trace of the complete modular exponentiation with message M and exponent d,
- $P(T_M, M, d_i)$ stand for the specific leakage pattern of a multiply done during the time frames T_M with exponent bit d_i,
- $P(T_S, M, d_i)$ stand for the specific leakage pattern of a squaring done during the time frames T_S with exponent bit d_i

With such definitions, $PCDC$ is expressed as follows:

$$PCDC(T_1, T_2) = \frac{\sigma_{(T_1)}}{\sigma_{(T_1 - T_2)}}. \quad (1)$$

In this expression, $\sigma_{(T_1)}$ denotes the standard deviation of the leakage vector T_1 while $\sigma_{(T_1 - T_2)}$ is the standard deviation of the difference $T_1 - T_2$. It should be noticed that $T_1 = P(T_S, M^2, d_{i-1})$ and $T_2 = P(T_S, M, d_i)$ if the

adversary aims at applying the DA. If he prefers the OA then $T_1 = P(T_S, -M, d_i)$ and $T_2 = P(T_S, M, d_i)$.

The idea on which the $PCDC$ is based is simple. If a collision occurs, $PCDC$ is expected to be really high because $\sigma_{(T_1-T_2)}$ is low while in the opposite case $PCDC$ is expected to be significantly smaller. If the results reported in [17] are convincing, demonstrating de facto the interest of this criterion, one should note that these results were obtained with averaged leakage traces, and it's not clear information of what means a low or a high $PCDC$ value is given. These latter points constitute important limitations and this especially as the $PCDC$ is an unbounded figure of merit.

To circumvent these issues, following the intuition of [17], we derived from equation 1 a criterion called Bounded Collision Detection Criterion afterward ($BCDC$), whose values are in]0,1]:

$$BCDC(T_1, T_2) = \frac{1}{\sqrt{2}} \times \frac{\sigma_{(T_1-T_2)}}{\sigma_{(T_1)}} \tag{2}$$

Property

Let us consider two patterns T_1 and T_2, we have:

1) a) Asymptotically

$$0 < BCDC(T_1, T_2) \leq 1 \tag{3}$$

b) Moreover in case of collision:

$$BCDC(T_1, T_2) \to 0, \tag{4}$$

when the SNR is high.

2) In the case of non-collision:

$$BCDC(T_1, T_2) = 1 \tag{5}$$

Proof

Each point of the pattern T_i (i=1,2), t_j^i (with $i \in [1, 2]$ and $j \in [1, w]$) is equal to the sum of :

- a deterministic value (s_j^i) representing the leakage of the DUT during the processing of the message.
- a random value representing the measurement noise (η_j^i) drawn in a normal distribution with mean 0 and standard deviation σ and independent of s_j^i.

Note that in the following we consider that the measurement noise is independent of leakage (s_j^i) in each point.
Considering:

$$\Delta T = T_1 - T_2 = [\delta_{t_1}, \cdots, \delta_{t_w}] \tag{6}$$

we can write each point:

$$\delta_{t_j} = t_j^1 - t_j^2 = s_j^1 + \eta_j^1 - (s_j^2 + \eta_j^2). \tag{7}$$

Considering the nature of the terms involved in equation 7, it appears that ΔT is a realization of a random variable with unknown mean but with a variance equal to:

$$\begin{aligned} \sigma^2_{(T_1-T_2)} = &\sigma^2([s_1^1 - s_1^2, \cdots, s_w^1 - s_w^2]) \\ &+ \sigma^2([\eta_1^1 - \eta_1^2, \cdots, \eta_w^1 - \eta_w^2]) \end{aligned} \tag{8}$$

which can be rewritten, defining σ_s as the standard deviation of $s_i = [s_1^i, \cdots, s_w^i]$, as follows:

$$\sigma_{(T_1-T_2)} = \sqrt{2 \cdot \sigma_s^2 + 2 \cdot \sigma^2} = \sqrt{2} \cdot \sqrt{\sigma_s^2 + \sigma^2} \tag{9}$$

1) From equation 9, one may express the asymptotic $BCDC$ values when a collision occurs or not.
In case of a collision, because
$t_j^1 = t_j^2$, and therefore because $\sigma_{(T_1-T_2)} = \sqrt{2} \cdot \sqrt{\sigma^2}$, equation 2 becomes:

$$\begin{aligned} BCDC(T_1, T_2) &= \frac{1}{\sqrt{2}} \frac{\sigma(\Delta T)}{\sigma(T_1)} \\ &= \frac{1}{\sqrt{2}} \frac{\sqrt{2}.\sigma}{\sqrt{\sigma_s^2 + \sigma^2}} \\ &= \frac{1}{\sqrt{1 + \frac{\sigma_s^2}{\sigma^2}}}, \end{aligned} \tag{10}$$

expression which tends towards 0 when the SNR is high ($\sigma_s >> \sigma$) or when pre-processing techniques are applied to increase the SNR value. It should be noticed that in our practical experiments (some examples are given in section IV-B), one shot leakage traces measured on a modern micro-controller are characterized by σ values close to that of σ_s resulting in $BCDC$ values close to $\frac{1}{2}$ in case of collisions.

2) In the opposite case, i.e. when no collision occurs, the expression of the $BCDC$ criterion is :

$$BCDC(T_1, T_2) = \frac{1}{\sqrt{2}} . \frac{\sqrt{2}.\sqrt{\sigma_s^2 + \sigma^2}}{\sqrt{\sigma_s^2 + \sigma^2}} = 1 \tag{11}$$

Here it should be noticed that all the above calculations do not take into account other electrical activities of the device under test. It should therefore be possible to observe some values slightly higher than 1 in practice. This could even occurs without any additional activity of the circuit because each pair of trace will give an estimation of $\sigma(T_1)$ and $\sigma(T_2)$ and therefore of the $BCDC$. The criterion should be considered as asymptotically bounded between 0 and 1.

C. Decision Making: collision or not

At that stage, we have at disposal a criterion that takes two leakage patterns and provides as a result a scalar value between $[0, 1]$. A BCDC equal to 0 means that a perfect collision occurs while a value of 1 means that there is no collision. However, 0 and 1 are asymptotic values and in practice $BCDC$ values range between 0 and 1. A decision tool is mandatory to automatically classify the obtained $BCDC$ values and thus decide who pairs of patterns are colliding.

At that end one can use the k-means algorithm (a standard clustering algorithm) with $k = 2$ in order to distinguish collision or not collision, as described by algorithm 3 and algorithm 4 for the DA and the OA respectively.

Algorithm 3 Collision detection method for Opposite Attack

Require: T_1^i: the $(i)^{th}$ pattern of M, T_2^i: the i^{th} pattern of $-Mmod\ n$, l:Number of operations
Ensure: Sequence of Modular Operations $S = [S_1, \cdots, S_l]$
1: **for** $i = 1\ \ to\ \ l$ **do**
2: $A_i = BCDC(T_1^i, T_2^i)$
3: **end for**
4: R = k-means $(A = [A_1, \cdots, A_l])$, with $k = 2$, $R_i \in \{N, C\}$
5: $S_0 = Square$
6: **for** $i = 1$ to length(R) **do**
7: **if** $(R_i == N)$ **and** $(S_{i-1}$ == Square$)$ **then**
8: S_i = Multiplication
9: **else**
10: S_i = Square
11: **end if**
12: **end for**
13: **return** S

Algorithm 4 Collision detection method for Doubling Attack

Require: $T_1^{(i+1)}$: the $(i+1)^{th}$ pattern of M, T_2^i: the i^{th} pattern of $M^2\ mod\ n$, l:Number of operations
Ensure: Sequence of Modular Operations $S = [S_1, \cdots, S_l]$
1: **for** $i = 1\ \ to\ \ l-1$ **do**
2: $A_i = BCDC(T_1^{(i+1)}, T_2^i)$
3: **end for**
4: R = k-means $(A = [A_1, \cdots, A_l])$, with $k = 2$, $R_i \in \{N, C\}$
5: $S_0 = Square$
6: **for** $i = 1$ to length(R) **do**
7: **if** $(R_{i-1} == N)$ **and** $(S_{i-1}$ == Square$)$ **then**
8: S_i = Multiplication
9: **else**
10: S_i = Square
11: **end if**
12: **end for**
13: **return** S

D. Enhancing SNR

In the preceding paragraphs, one could have noticed the importance of the measurement quality, i.e. the SNR, on the $BCDC$ values in case of a collision. It is therefore interesting to take a look to practical techniques for enhancing the SNR. Among the methods targeting the SNR improvement we have chosen filtering and trace compression. The averaging technique has not been considered because not applicable due to misalignment of traces. This technique is indeed not applicable in many cases due to misalignment of traces or again to the more and more common usage of multiple clocks within smart-cards.

1) Filtering: Filtering techniques are efficient if one knows which frequency bandwidths must be kept and which must be attenuated during in the considered side channel. In [21] a method has been proposed to select which frequencies must be kept or not during SCA. This method is based on a selection criterion called the Leakage to Noise Ratio (LNR). When leakage traces are power measurements, its expression is:

$$LNR(f) = \frac{1}{f^2}\frac{\langle PSD(f)\rangle}{\sigma_{PSD(f)}} \tag{12}$$

where $< PSD(f) >$ is the mean Power Spectral Density at the frequency f and $\sigma_{PSD(f)}$ the standard deviation of PSD(f). This technique can be used to enhance the detection of collisions as it will shown in section IV-B.

2) Trace compression: Another method would to apply the compression method on both curves. In this work we have chosen the compression method which consists in summing each set of c consecutive samples into one to obtain a compressed trace. Choosing a suitable value for c is therefore crucial for the efficiency of this compression technique. In practice, usually time intervals up to the length of a single clock cycle are used. These settings given, let us show why and how the compression enhances the contrast between collisions and no collision when the adversary uses the $BCDC$.

Compressing a leakage trace T_i by a factor c results in the getting of the compressed trace:

$$C^c(T_i) = \left\{ t'^i_l = \sum_{k=lc}^{(l+1)c-1} t^i_k, l \in [0, w/c] \cap \mathbf{N} \right\} \tag{13}$$

Assuming the realizations of the noise are independent from one sample to another, one may consider the central limit theorem and express the standard deviation of the compressed trace:

$$\sigma_{C^c(T_i)} = \sqrt{\sigma'^2_s + \frac{\sigma^2}{c}} \tag{14}$$

where σ'_s is the standard deviation of compressed deterministic part s^i_j of T_i on which the central limit theorem can not be applied because the computations are deterministic and therefore independent. As shown, compressing by a factor c the leakage trace T_i results in dividing the standard deviation of the noise by $\frac{1}{\sqrt{c}}$. This reduction does not change the value of the $BCDC$ when there is no collision because the reduction of the noise standard deviation is involved in both the numerator and in the denominator of equation 11). However, when there is a collision compressing by a factor c results in dividing significantly the value of the $BCDC$. One may therefore conclude that compressing traces allows increasing the contrast between the $BCDC$ values associated to the occurrence or not of a collision and therefore helps the k-means in partitioning accurately collisions for non-collisions.

IV. Experimental Results

This section gives some of the results obtained during many experimental campaigns. The results we have chosen to report aims at sustaining the theoretical efficiency of our automated collision detection technique which is based on the joint usage of the $BCDC$ and of the k-means. Some results showing the efficiency of the LNR but also of the compression technique are also given.

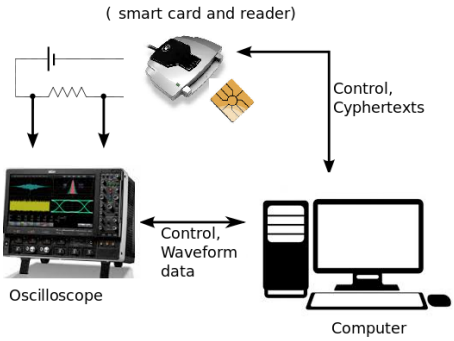


Fig. 4: Power Analysis materials

A. Device Under Test and measurement platform

This subsection present results obtained on different products (DUTs). The DUT references are kept confidential even if these products can be purchase on the web. All DUTs are open development platforms where it is possible to call the Crypto routines with our own inputs (Messages). To test the efficiency of CBA but also of our detection technique, we performed many analysis using the platform shown Figure 4. This platform features a LeCroy TELEDYNE HD04096 oscilloscope to collect the power consumption traces. The computer sends instructions and data to a modern micro-controller interface, and receive the output data.

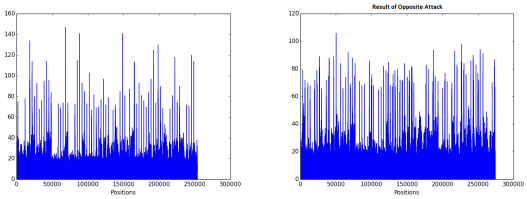
During all experiments, the DUT was executing an Exponentiation with the exponent value $d = AE8F8A89632F1EE49A13548690183427$. Only one shot traces were collected during our experiments considering that exponent randomization is now a standard countermeasure. Traces were collected using a sampling rate of 2.5 GS/s (or 500 MS/s) resulting in patterns of 24000 samples (or 4850 respectively).

B. Experimental results

1) Setting a reference using the Difference of Traces (DoT): To evaluate the efficiency of the detection technique introduced above, we first set an efficiency reference. This reference corresponds to the capability of detecting collision using only the difference between the two traces (DoT) i.e. using the solution introduced in the seminal works about CBA [7], [24].

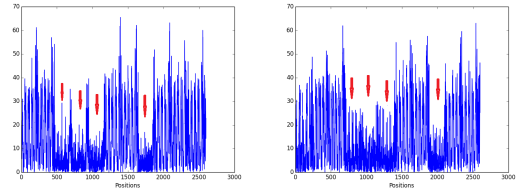
We therefore collected pairs of traces corresponding to the processing of M and $M^2 \ mod \ n$ or to that of M and $-M \ mod \ n$, for several values of M selected randomly and we searched for collisions on the third byte ($8A = 10001010_2$). After alignment of traces and patterns with the POC method, we computed the DoT directly to draw Figure 5a and 5b. As shown, even if the difference is low on some time frames, it is difficult to detect with a high level of confidence any collision.

2) Efficiency of the compression technique: The reference being set, we applied on the same traces a compression of factor $c = 100$ and then recomputed the difference. Figures 6a and 6b show the results of the DA and the OA. These figures should be compared to Figures 5a and 5b respectively. As shown, visual inspection allows guessing some collisions marked by red arrows. Compression clearly enhances the results. However, at that stage, visual inspection remains the decision tool for identifying where collisions occur.



(a) Doubling Attack (b) Opposite Attack

Fig. 5: Absolute value of the DoT obtained when applying the CBA on the third byte of the exponent

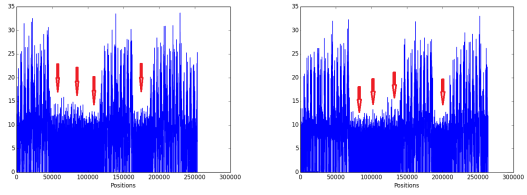


(a) Doubling Attack (b) Opposite Attack

Fig. 6: Absolute value of the DoT obtaind when applying the CBA on the third byte of the exponent after compression ($c = 100$) of traces

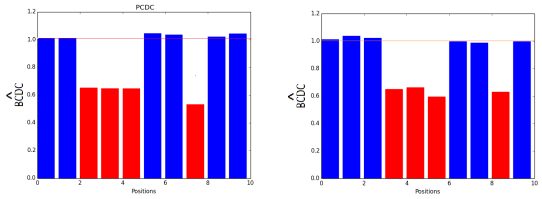
3) Efficiency of the LNR filtering: The same experiment than in the preceding section was conducted using the LNR instead of the compression technique. Figures 7a and 7b give the results for the DA and OA. As for Figures 5a and 5b, collisions can be detected by visual inspection. However, the contrast between collision and no collision is not as high than when using compression.

4) Efficiency of the BCDC: In this paragraph some results corresponding to the direct application of our detection technique, which is based on the joint usage of the $BCDC$ and of the k-means, are given. It should be noticed that no technique to enhance the SNR was used to obtained these results.



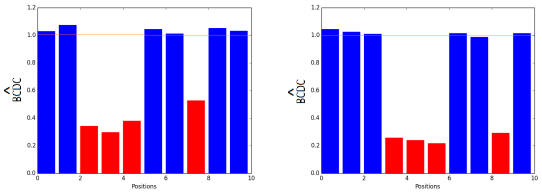
(a) Doubling Attack (b) Opposite Attack

Fig. 7: Absolute value of the DoT obtained when applying the CBA on the third byte of the exponent after LNR filtering of traces



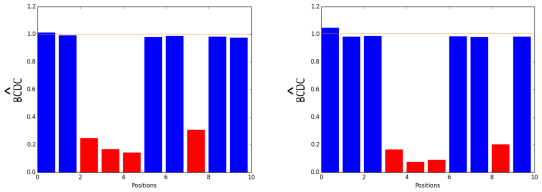
(a) Doubling Attack (b) Opposite Attack

Fig. 8: Results obtained by direct application of the proposed detection technique (a,b) the BCDC and k-means on the third byte of the exponent



(a) Doubling Attack (b) Opposite Attack

Fig. 9: Results obtained by combining the proposed detection technique (BCDC and k-means) and of compression ($c = 100$)



(a) Doubling Attack (b) Opposite Attack

Fig. 10: Results obtained by combining the proposed detection technique (BCDC and k-means) and LNR filtering

Figures 8a and 8b show the results of the two considered attacks applied on raw traces (no compression nor filtering). In these figures, each bar represents a BCDC of two adjacent modular operations and its color shows its membership to the cluster of collision (red)or to that of non collision (blue). Comparing figures 5a and 5b with figures 8a and 8b highlights a first key advantage of using the $BCDC$ which is the association of a scalar value to each modular operation. In addition the BCDC eases the detection of collisions by simple visual inspection because its values are bounded and the bounds are easily interpretable ('1' no collision, '0' collision).

One can also remark in figure 8 that $BCDC$ values associated to no collision are close to one as expected from the calculation given in section III. One can also notice that $BCDC$ values in case of collision range between 0.6 and 0.8. Collisions can easily be distinguished from non-collision using the k-means. An adversary can therefore easily recover the sequence of operations. In Figure 8a, he obtains the following sequence $NNCCCNNCNN$, where N means non-collision, C collision. Applying algorithm 4 finally provides the exponents bits $(1000101x)_2$ where x is the LSB of the Byte to recover.

5) Combining BCDC and compression: If the results introduced in the preceding paragraphs are sufficient to validate the correctness of the proposed detection criterion, we can combining BCDC and Compression, once again on the same measurements, after application of a compression by a factor $c = 100$. The comparison of figures 9a and 9b with figures 8a and 8b respectively clearly shows that the usage of the compression technique significantly increases the contrast between collision and no-collision. The $BCDC$ values in case of collision are now ranging between 0.2 and 0.4.

6) Combining BCDC and LNR filtering: All the experiments described in the preceding section were re-done using LNR filtering rather than compression. Figures 10a and 10b give the results for the DA and OA respectively. As expected from [21], giving more importance to low frequencies than to high frequencies allows increasing significantly the SNR.

7) Overall efficiency estimation (Success Rate): To definitively validate the interest of the proposed detection technique ($BCDC$ and k-means), we estimated the probability to detect a collision knowing the exponent. This was done by considering a set of 100 pairs of traces for the DA and the OA. This probability was called Success Rate by analogy to the current practices while dealing with vertical SCA. It should be also noticed that our detection technique (a criterion and the k-means) was applied using both the $PCDC$ and the $BCDC$, using compression or not, or LNR filtering or not. Table I and Table II lists the results obtained for different factors of compression c but also for two different sampling rate values.

As shown, the application of compression or LNR filtering has a moderated effect on the Success Rate value when the scope was set with a high sampling rate value (here 2.5 GS/s) to collect traces. It should be noticed that high sampling rate values can not be used for long secret exponents because of memory limitations of oscilloscopes. One can also observe that compressing too much the leakage traces could be counterproductive. One may therefore conclude that the usage of the $BCDC$ coupled with that of the k-means is sufficient to automatically detect collisions with a Success Rate values higher than 99% when leakage traces are acquired with a sampling rate.

Table II also shows that when the sampling rate is low, LNR filtering and compression enhance the results and that the Success Rate reaches 100%. However, it should be noticed once again than compressing too much the leakage traces is counterproductive.

V. Conclusion

In this paper we have introduced a new evaluation collision criterion called BCDC. This criterion has been proposed to first make collision attack much practical. Some experiments are presented to compare this criterion with the already published in [15]. As a clear result our new criterion is much more efficient to find and exploit collisions using Collision Based

Sampling Rate	2.5 GS/s			
Method	Doubling Attack		Opposite Attack	
	PCDC	BCDC	PCDC	BCDC
No Trace pre-processing	90%	97%	91%	**99%**
$c = 10$	95%	97%	**98%**	**100%**
$c = 20$	96%	96%	92%	94%
$c = 50$	92%	**98%**	88%	90%
$c = 100$	83%	**98%**	79%	97%
$c = 200$	53%	91%	34%	68%
LNR	93%	94%	70%	**99%**

TABLE I: Success Rate using the detection technique with the $PCDC$ and $BCDC$ on traces collected with a sampling rate equal to 2.5 GS/s

Sampling Rate	500 MS/s			
Method	Doubling Attack		Opposite Attack	
	PCDC	BCDC	PCDC	BCDC
No Trace pre-processing	87%	**99%**	75%	97%
$c = 10$	92%	**100%**	90%	**100%**
$c = 20$	90%	**100%**	90%	**100%**
$c = 50$	74%	**100%**	71%	**100%**
$c = 100$	68%	96%	68%	85%
$c = 200$	67%	74%	61%	70%
LNR	82%	**100%**	80%	**100%**

TABLE II: Success Rate using the detection technique with the $PCDC$ and $BCDC$ on traces collected with a sampling rate equal to 500 MS/s

Attack (CBA) from the literature [7], [24], [23], [9], [10], [20]. Even if the result presented in the paper are focusing on modular exponentiation [7], [24] our criteria is also effective on Symmetric Cryptosystems. In addition some pre-processing techniques has been investigated (Filtering and Compression) with both a theoretical and a practical statement showing concrete elements. The presented methodology based on both the new criterion coupled with a k-means processing shown as demonstrated, determinant practical way to exploit collisions.

REFERENCES

[1] Chen Aidong, Xu Sen, Chen Yun, and Qin Zhiguang. Collision-based chosen-message simple power clustering attack algorithm. *Communications, China*, 10(5):114–119, 2013.

[2] Aurélie Bauer, Eliane Jaulmes, Emmanuel Prouff, and Justine Wild. Horizontal and vertical side-channel attacks against secure rsa implementations. In *Topics in Cryptology–CT-RSA 2013*, pages 1–17. Springer, 2013.

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

[4] Christophe Clavier, Benoit Feix, Georges Gagnerot, Christophe Giraud, Mylene Roussellet, and Vincent Verneuil. Rosetta for single trace analysis. In *Progress in Cryptology-INDOCRYPT 2012*, pages 140–155. Springer, 2012.

[5] Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Horizontal correlation analysis on exponentiation. In *Information and Communications Security*, pages 46–61. Springer, 2010.

[6] Jean-Sébastien Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In *Cryptographic Hardware and Embedded Systems, First International Workshop, CHES'99, Worcester, MA, USA, August 12-13, 1999, Proceedings*, volume 1717 of *Lecture Notes in Computer Science*, pages 292–302. Springer, 1999.

[7] P. A. Fouque and F. Valette. The doubling attack - why upwards is better than downwards. In *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, volume 2779 of *Lecture Notes in Computer Science*, pages 269–280. Springer, 2003.

[8] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.

[9] Naofumi Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, and Adi Samir. Comparative power analysis of modular exponentiation algorithms. *Computers, IEEE Transactions on*, 59(6):795–807, 2010.

[10] Naofumi Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, and Adi Shamir. Collision-based power analysis of modular exponentiation using chosen-message pairs. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop, Washington, D.C., USA, August 10-13, 2008. Proceedings*, volume 5154 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2008.

[11] Naofumi Homma, Sei Nagashima, Yuichi Imai, Takafumi Aoki, and Akashi Satoh. High-resolution side-channel attack using phase-based waveform matching. In *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop*, volume 4249 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2006.

[12] Cetin Kaya Koc. High-speed rsa implementation. Technical report, Technical Report, RSA Laboratories, 1994.

[13] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[14] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[15] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*, volume 31. Springer Science & Business Media, 2008.

[16] Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[17] Guilherme Perin, Laurent Imbert, Lionel Torres, and Philippe Maurine. Practical analysis of rsa countermeasures against side-channel electromagnetic attacks. In *Smart Card Research and Advanced Applications*, pages 200–215. Springer, 2014.

[18] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[19] Kai Schramm, Thomas J. Wollinger, and Christof Paar. A new class of collision attacks and its application to des. In *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 206–222. Springer, 2003.

[20] YEN Sung-Ming, LIEN Wei-Chih, and CHEN Chien-Ning. Modified doubling attack by exploiting chosen ciphertext of small order. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 94(10):1981–1990, 2011.

[21] Sébastien Tiran, Sébastien Ordas, Yannick Teglia, Michel Agoyan, and Philippe Maurine. A model of the leakage in the frequency domain and its application to cpa and dpa. *Journal of Cryptographic Engineering*, 4(3):197–212, 2014.

[22] Colin D. Walter. Sliding windows succumbs to big mac attack. In *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 286–299. Springer, 2001.

[23] Sung-Ming Yen, Lee-Chun Ko, SangJae Moon, and JaeCheol Ha. Relative doubling attack against montgomery ladder. In *Information Security and Cryptology-ICISC 2005*, pages 117–128. Springer, 2006.

[24] Sung-Ming Yen, Wei-Chih Lien, SangJae Moon, and JaeCheol Ha. Power analysis by exploiting chosen message and internal collisions–vulnerability of checking mechanism for rsa-decryption. In *Progress in Cryptology–Mycrypt 2005*, pages 183–195. Springer, 2005.