Horizontal Collision Correlation Attack on Elliptic Curves

Aurélie Bauer(✉), Eliane Jaulmes, Emmanuel Prouff, and Justine Wild

ANSSI, 51, Bd de la Tour-Maubourg, SP 07, 75700 Paris, France
{aurelie.bauer,eliane.jaulmes,emmanuel.prouff,justine.wild}@ssi.gouv.fr

Abstract. Elliptic curves based algorithms are nowadays widely spread among embedded systems. They indeed have the double advantage of providing efficient implementations with short certificates and of being relatively easy to secure against side-channel attacks. As a matter of fact, when an algorithm with constant execution flow is implemented together with randomization techniques, the obtained design usually thwarts classical side-channel attacks while keeping good performances. Recently, a new technique that makes some randomizations ineffective, has been successfully applied in the context of RSA implementations. This method, related to a so-called *horizontal modus operandi*, introduced by Walter in 2001, turns out to be very powerful since it only requires leakages on a single algorithm execution. In this paper, we combine such kind of techniques together with the *collision correlation* analysis, introduced at CHES 2010 by Moradi *et al.*, to propose a new attack on elliptic curves atomic implementations (or unified formulas) with input randomization. We show how it may be applied against several state-of-the art implementations, including those of Chevallier-Mames *et al.*, of Longa and of Giraud-Verneuil and also Bernstein and Lange for unified Edward's formulas. Finally, we provide simulation results for several sizes of elliptic curves on different hardware architectures. These results, which turn out to be the very first horizontal attacks on elliptic curves, open new perspectives in securing such implementations. Indeed, this paper shows that two of the main existing countermeasures for elliptic curve implementations become irrelevant when going from vertical to horizontal analysis.

1 Introduction

Elliptic Curves Cryptosystems (ECC) that have been introduced by N. Koblitz [21] and V. Miller [29], are based on the notable *discrete logarithm problem*, which has been thoroughly studied in the literature and is supposed to be a hard mathematical problem. The main benefit in elliptic curves based algorithms is the size of the keys. Indeed, for the same level of security, the schemes require keys that are far smaller than those involved in classical public-key cryptosystems. The success of ECC led to a wide variety of applications in our daily life and they are now implemented on lots of embedded devices: smart-cards, micro-controller,

T. Lange, K. Lauter, and P. Lisoněk (Eds.): SAC 2013, LNCS 8282, pp. 553–570, 2014.
DOI: 10.1007/978-3-662-43414-7_28, © Springer-Verlag Berlin Heidelberg 2014

and so on. Such devices are small, widespread and in the hands of end-users. Thus the range of threats they are confronted to is considerably wider than in the classical situation. In particular, physical attacks are taken into account when assessing the security of the application implementation (e.g. the PACE protocol in e-passports [20]) and countermeasures are implemented alongside the algorithms.

A physical attack may belong to one of the two following families: *perturbation analysis* or *observation analysis*. The first one tends to modify the cryptosystem processing with laser beams, clock jitter or voltage perturbation. Such attacks can be thwarted by monitoring the device environment with captors and by verifying the computations before returning the output. The second kind of attacks consists in measuring a physical information, such as the power consumption or the electro-magnetic emanation, during sensitive computations. Inside this latter area we can distinguish, what we call *simple attacks*, that directly deduces the value of the secret from one or a small number of observation(s) (e.g. *Simple Power Analysis* [23]) and *advanced attacks* involving a large number of observations and exploiting them through statistics (e.g. *Differential Power Analysis* [24] or *Correlation Power Analysis* [9]). Such attacks require the use of a statistical tool, also known as a *distinguisher*, together with a *leakage model* to compare hypotheses with real traces (each one related to known or chosen inputs). The latter constraint may however be relaxed thanks to the so-called *collision attacks* [32] which aim at detecting the occurrences of colliding values during a computation, that can be linked to the secret [8,14,30,31]. In order to counteract all those attacks, randomization techniques can be implemented (e.g. scalar/message blinding for ECC [16]). The recent introduction of the so-called *horizontal* side-channel technique by Clavier *et al.* in [13] seems to have set up a new deal. This method, which is inspired by Walter's work [33], takes its advantage in requiring a unique power trace, thus making classical randomization techniques ineffective. Up to now, it has been applied successfully on RSA implementations and we show in this paper that it can be combined with collision correlation analysis to provide efficient attack on elliptic curves protected implementations.

Core idea. In the context of embedded security, most ECC protocols (e.g. ECDSA [1] or ECDH [2]) use a short term secret that changes at each protocol iteration. In this particular setting, advanced side-channel attacks, which require several executions of the algorithm with the same secret, are ineffective. As a consequence, only protection against SPA is usually needed, that can be done thanks to the popular *atomicity* principle [11,18,26]. Up to now, this technique is considered as achieving the best security/efficiency trade-off to protect against side-channel analysis. In this paper, we provide a new side-channel attack, called *horizontal collision correlation analysis* that defeats such protected ECC implementations. In particular, implementations using point/scalar randomization combined with atomicity are not secure, contrary to what was thought up to now. Moreover in order to complete our study, we also investigate the case of unified formulas[1]. Indeed, we show that our horizontal collision correlation attack allows to distinguish, with a single leakage trace, a doubling operation from an addition one. This technique, which allows to eventually recover the secret scalar, is applied to three different atomic formulae on elliptic curves, namely those proposed by Chevallier-Mames *et al.* in [11], by Longa in [26], by Giraud and Verneuil in [18].

The paper is organized as follows. First, Sect. 2 recalls some basics about ECC in a side-channel attacks context. Then, under the assumption that one can distinguish common operands in modular multiplications, the outlines of our new *horizontal collision correlation* attack are presented in Sect. 3. After a theoretical analysis explaining how to practically deal with the distinguishability assumption, we provide in Sect. 4 experimental results for 160, 256 and 384-bit-size curves working with 8, 16 or 32-bit registers. These results show that the attack success rate stays high even when significant noise is added to the leakage.

2 Preliminaries

2.1 Notations and Basics on Side-Channel Attacks

Notations. A realization of a random variable X is referred to as the corresponding lower-case letter x. A *sample* of n observations of X is denoted by (x) or by $(x_i)_{1\leq i\leq n}$ when a reference to the indexation is needed. In this case, the global event is summed up as $(x) \hookleftarrow X$. The j^{th}coordinate of a variable X (resp. a realization x), viewed as a vector, is denoted by $X[j]$ (resp. $x[j]$). As usual, the notation $\mathbb{E}[X]$ refers to the mean of X. For clarity reasons we sometimes use the notation $\mathbb{E}_X[Y]$ when Y depends on X and other variables, to enlighten the fact that the mean is computed over X. Attacks presented in this paper involve the *linear correlation coefficient* which measures the linear interdependence between two variables X and Y. It is defined as $\rho(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X\sigma_Y}$, where $\text{cov}(X,Y)$, called *covariance between* X and Y, equals $\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$ and where σ_X and σ_Y respectively denotes the standard deviation of X and Y. The linear correlation coefficient can be approximated from realizations samples $(x_i)_{1\leq i\leq n}$ and $(y_i)_{1\leq i\leq n}$ of X and Y respectively. For this approximation, the following so-called *Pearson's coefficient* is usually involved:

$$\hat{\rho}(X,Y) = \frac{n\sum_i x_i y_i - \sum_i x_i \sum_j y_j}{\sqrt{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2}\sqrt{n\sum_j y_j^2 - \left(\sum_j y_j\right)^2}} . \tag{1}$$

General Attack Context. In the subsequent descriptions of side-channel analyses, an algorithm $\mathcal{A}$ is modelled by a sequence of *elementary calculations* $(\mathsf{C}_i)_i$ that are Turing machines augmented with a common random access

[1] Among the unified formulas, we especially focus on the Edward's ones in [5] introduced by Bernstein and Lange since they lead to efficient doubling and addition computations compared to the Weierstrass case [10].

memory (see [28] for more details about this model). Each elementary calculation C_i reads its input X_i in this memory and updates it with its output $O_i = \mathsf{C}_i(X_i)$. During the processing of $\mathcal{A}$, each calculation C_i may be associated with an information *leakage* random variable L_i (a.k.a. *noisy observation*). A prerequisite for the side-channel analyses described in this paper to be applicable is that the *mutual information* between O_i and L_i is non-zero. The alternative notation $L_i(O_i)$ will sometimes be used to stress the relationship between the two variables.

A side-channel analysis aims at describing a strategy to deduce information on the algorithm secret parameter from the leakages L_i. Let us denote by $\boldsymbol{s}$ this secret parameter. In this paper, we pay particular attention to two attacks sub-classes. The first ones are called *simple* and try to exploit a dependency between the sequence of operations C_i and $\boldsymbol{s}$ (independently of the C_i inputs and outputs). A well-known example of such an attack is the simple power analysis (SPA) [16]. In this attack, the algorithm input is kept constant and the unprotected sequence of C_i is usually composed of two distinct operations (for instance a doubling and an addition in the case of ECC). It can easily be checked that the order of those operations in the sequence is a one-to-one function of the secret scalar $\boldsymbol{s}$. Hence, if the leakages L_i enable to clearly differentiate the operations, then the adversary may recover the order of the latters, and thus the secret.

Following the framework presented in [4], we call *advanced* the attacks belonging to the second class of side-channel analyses. Among them, we find the well-known differential power analysis (DPA) [24] or the correlation power analysis (CPA) [9]. Contrary to simple attacks, the advanced ones do not only focus on the *operations* but also on the *operands*. They usually focus on a small subset I of the calculations C_i and try to exploit a statistical dependency between the results O_i of those calculations and the secret $\boldsymbol{s}$. For such a purpose, the adversary must get a sufficiently large number N of observations $(\ell_j^i)_j \hookleftarrow L_i(O_i)$, where $i \in I$ and $1 \leq j \leq N$.

In the literature, two strategies have been specified to get the observations samples $(\ell_j^i)_j$ for a given elementary computation $O_i = \mathsf{C}_i(X_i)$. The first method, called *vertical*, simply consists in executing the implementation several times and in defining ℓ_j^i as the observation related to the result O_i at the j^{th}algorithm execution. Most attacks [3,9,24] enter into this category and the number of different indices i may for instance correspond to the *attack order* [27]. The second method, called *horizontal* [13,33], applies on a single algorithm execution. It starts by finding the sequence of elementary calculations $(\mathsf{C}_{i_j})_j$ that processes the same mathematical operation than C_i (e.g. a field multiplication) and depends on the same secret sub-part. By construction, all the outputs O_{i_j} of the C_{i_j} can be viewed as a realization of $O_i = \mathsf{C}_i(X_i)$ and the ℓ_j^i are here defined as the observations of the O_{i_j}. We can eventually notice that the vertical and horizontal strategies are perfectly analogous to each other and that they can be applied to both simple and advanced attacks.

2.2 Background on Elliptic Curves

As this paper focuses on side-channel attacks on ECC, let us recall now some basics on elliptic curves and especially on the various ways of representing points on such objects (the reader could refer to [15,19] for more details).

Throughout this paper, we are interested in elliptic curve implementations running on platforms (ASIC, FPGA, micro-controller) embedding a hardware modular multiplier (e.g. a 16-bit, 32-bit or 64-bit multiplier). On such implementations, the considered elliptic curves are usually defined over a prime finite field $\mathbb{F}_p$. In the rest of this paper, we will assume that all curves are defined over $\mathbb{F}_p$ with $p \neq \{2, 3\}$. The algorithm used for the hardware modular multiplication is assumed to be known to the attacker. Moreover, to simplify the attack descriptions, we assume hereafter that the latter multiplication is performed in a very simple way: a schoolbook long integer multiplication followed by a reduction. Most of current devices do not implement the modular multiplications that way, but the attacks described hereafter can always be adapted by changing the definition of the elementary operations of Sect. 3.3 (see the full version of the paper for a complete discussion on that point).

Definition. An elliptic curve E over a prime finite field $\mathbb{F}_p$ with $p \neq \{2, 3\}$ can be defined as an algebraic curve of affine reduced Weierstrass equation:

$$(E) : y^2 = x^3 + ax + b \ , \tag{2}$$

with $(a, b) \in (\mathbb{F}_p)^2$ and $4a^3 + 27b^2 \neq 0$. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on (E), the sum $R = (x_3, y_3)$ of P and Q belongs to the curve under a well-known addition rule [21]. The set of pairs $(x, y) \in (\mathbb{F}_p)^2$ belonging to (E), taken with an extra point $\mathcal{O}$, called *point at infinity*, form an abelian group named $E(\mathbb{F}_p)$.

In the rest of the paper, the points will be represented using their projective coordinates. Namely, a point $P = (x, y)$ is expressed as a triplet $(X : Y : Z)$ such that $X = xZ$ and $Y = yZ$.

2.3 Points Operations in Presence of SCA

This paper focusses on elliptic curves cryptosystems which involve the scalar multiplication $[\boldsymbol{s}]P$, implemented with the well-known *double and add* algorithm.

In a non-protected implementation, the sequence of point doublings and point additions can reveal the value of $\boldsymbol{s}$ with a single leakage trace. Thus to protect the scheme against SPA, the sequence of point operations must be independent from the secret value. This can be achieved in several ways. The double and add *always* algorithm [16] is the simplest solution. It consists in inserting dummy point additions each time the considered bit value of $\boldsymbol{s}$ is equal to 0. In average, this solution adds an overhead of $\frac{\log_2(\boldsymbol{s})}{2}$ point additions. Another technique consists in using unified formulae for both addition and doubling [6,7,25]. Finally, the scheme that is usually adopted in constrained devices such as smart cards, since it achieves the best time/memory trade-off, remains atomicity [11,18,26].

This principle is a refinement of the double and add always technique. It consists in writing addition and doubling operations as a sequence of a unique pattern. This pattern is itself a sequence of operations over $\mathbb{F}_p$. Since the pattern is unique, the same sequence of field operations is repeated for the addition and the doubling, the only difference being the number of times the pattern is applied for each operation. It thus becomes impossible to distinguish one operation from the other or even to identify the starting and ending of these operations.

To defeat an atomic implementation, the adversary needs to use advanced side-channel attacks (see Sect. 2.1), such as DPA, CPA and so on. These attacks focus on the operations operands instead of only focusing on the kind of operations. They usually require more observations than for SPA since they rely on statistical analyses. In the ECC literature, such attacks have only been investigated in the vertical setting, where they can be efficiently prevented by input randomization.

3 Horizontal Collision Correlation Attack on ECC

We show hereafter that implementations combining atomicity and randomization techniques are in fact vulnerable to collision attacks in the horizontal setting. This raises the need for new dedicated countermeasures.

This section starts by recalling some basics on collision attacks. Then, assuming that the adversary is able to distinguish when two field multiplications have a common (possibly unknown) operand, we show how to exhibit flaws in the atomic algorithms proposed in [11,18,26]) and also in implementations using the unified formulas for Edward's curves [5]. Eventually, we apply the collision attack presented in the first subsection to show how to efficiently deal with the previous assumption.

3.1 Collision Power Analysis in the Horizontal Setting

To recover information on a subpart s of the secret $\boldsymbol{s}$, collision side-channel analyses are usually performed on a sample of observations related to the processing, by the device, of two variables O_1 and O_2 that jointly depend on s. The advantage of those attacks, compared to the classical ones, is that the algorithm inputs can be unknown since the adversary does not need to compute predictions on the manipulated data. When performed in the horizontal setting, the observations on O_1 and O_2 are extracted from the same algorithm execution (see Sect. 2.1). Then, the correlation between the two samples of observations is estimated thanks to the Pearson's coefficient (see Eq. (1)) in order to recover information on s. We sum up hereafter the outlines of this attack, that will be applied in the following.

Remark 1. In Table 1, we use Pearson's coefficient to compare the two samples of observations but other choices are possible (e.g. mutual information).

Remark 2. In order to deduce information on s from the knowledge of $\hat{\rho}$, one may use for instance a *Maximum Likelihood* distinguisher (see a discussion on that point in Sect. 4).

Table 1. Collision power analysis

1. Identify two elementary calculations $\mathsf{C}_1(\cdot)$ and $\mathsf{C}_2(\cdot)$ which are processed several times, say N, with input(s) drawn from the same distribution(s). The correlation between the random variables O_1 and O_2 corresponding to the outputs of C_1 and C_2 must depend on the same secret sub-part s.
2. For each of the N processings of C_1 (resp. C_2) get an observation ℓ_j^1 (resp. ℓ_j^2) with $j \in [1; N]$.
3. Compute the quantity: $\hat{\rho} = \hat{\rho}\Big((\ell_j^1)_j, (\ell_j^2)_j\Big)$
4. Deduce information on s from $\hat{\rho}$.

In the next section, the attack in Table 1 is invoked as an Oracle enabling to detect whether two field multiplications share a common operand.

Assumption 1. *The adversary can detect when two field multiplications have at least one operand in common.*

In Sect. 3.3, we will come back to the latter hypothesis and will detail how it can indeed be satisfied in the particular context of ECC implementations on constrained systems.

3.2 Attacks on ECC Implementations: Core Idea

We start by presenting the principle of the attack on atomic implementations, and then on an implementation based on unified (addition and doubling) formulas over Edward's curves.

Attack on Chevallier-Mames* et al.*'s Scheme. In Chevallier-Mames *et al.*'s atomic scheme, historically the first one, the authors propose the three first patterns[2] given in Fig. 1 for the doubling of a point $Q = (X_1 : Y_1 : Z_1)$ and the addition of Q with a second point $P = (X_2 : Y_2 : Z_2)$.

As expected, and as a straightforward implication of the atomicity principle, the doubling and addition schemes perform exactly the same sequence of field operations if the *star* (dummy) operations are well chosen[3]. This implies that it is impossible to distinguish a doubling from an addition by just looking at the sequence of calculations (i.e. by SPA). Let us now focus on the operations' operands. In the addition scheme, the field multiplications in Patterns 1 and 3 both involve the coordinate Z_2. On the contrary, the corresponding multiplications in the doubling scheme have *a priori* independent operands (indeed the first one corresponds to the multiplication $X_1 \cdot X_1$, whereas the other one corresponds to $Z_1^2 \cdot Z_1^2$). If an adversary has a mean to detect this difference (which is actually the case under Assumption 1), then he is able to distinguish a doubling from an addition and thus to fully recover the secret scalar $\boldsymbol{s}$. Indeed, let us

[2] For readability reasons we do not recall the full patterns but the interested reader can find them in [11].

[3] Guidelines are given in [11] to define the dummy operations in a pertinent way.

	Doubling		Addition
	$R_0 \leftarrow a$, $R_1 \leftarrow X_1$, $R_2 \leftarrow Y_1$, $R_3 \leftarrow Z_1$		$R_1 \leftarrow X_1$, $R_2 \leftarrow Y_1$, $R_3 \leftarrow Z_1$, $R_7 \leftarrow X_2$, $R_8 \leftarrow Y_2$, $R_9 \leftarrow Z_2$
1.	$\boldsymbol{R_4 \leftarrow R_1 \cdot R_1}$ $(= X_1 \cdot \boldsymbol{X_1})$ $R_5 \leftarrow R_4 + R_4$ $\star$ $R_4 \leftarrow R_4 + R_5$	1.	$\boldsymbol{R_4 \leftarrow R_9 \cdot R_9}$ $(= Z_2 \cdot \boldsymbol{Z_2})$ $\star$ $\star$ $\star$
2.	$R_5 \leftarrow R_3 \cdot R_3$ $R_1 \leftarrow R_1 + R_1$ $\star$ $\star$	2.	$R_1 \leftarrow R_1 \cdot R_4$ $\star$ $\star$ $\star$
3.	$\boldsymbol{R_5 \leftarrow R_5 \cdot R_5}$ $(= Z_1^2 \cdot \boldsymbol{Z_1^2})$ $\star$ $\star$ $\star$	3.	$\boldsymbol{R_4 \leftarrow R_4 \cdot R_9}$ $(= Z_2^2 \cdot \boldsymbol{Z_2})$ $\star$ $\star$ $\star$

Fig. 1. Three first atomic patterns of point doubling and addition.

focus on the processing of the second step of the double and add left-to-right algorithm, and let us denote by s the most significant bit of $\boldsymbol{s}$. Depending on s, this sequence either corresponds to the processing of the doubling of $Q = [2]P$ (case $s = 0$) or to the addition of $Q = [2]P$ with P (case $s = 1$). Eventually, the results T_1 and T_2 of the field multiplications in respectively Patterns 1 and 3 satisfy:

$$\begin{cases} T_1 = \left(X_1 \cdot X_1\right)^{1-s} \cdot \left(Z_2 \cdot Z_2\right)^{s} \\ T_2 = \left(Z_1^2 \cdot Z_1^2\right)^{1-s} \cdot \left(Z_2^2 \cdot Z_2\right)^{s} \end{cases}, \tag{3}$$

where we recall that we have $P = (X_2 : Y_2 : Z_2)$ and $Q = (X_1 : Y_1 : Z_1)$. Equation (3) and Assumption 1 enables to deduce whether s equals 0 or 1. Applying this attack $\log_2(\boldsymbol{s})$ times, all the bits of $\boldsymbol{s}$ can be recovered one after the other.

We now show that the same idea can successfully be applied to attack the other atomic implementations proposed in the literature, namely those of Longa [26] and Giraud and Verneuil [18].

Attack on Longa's Scheme. The atomic pattern introduced by Longa in [26] is more efficient than that of Chevallier-Mames *et al.*'s scheme. This improvement is got by combining affine and Jacobian coordinates in the points addition, see Fig. 2.

It can be seen that the first and third patterns of Longa's scheme contain two field multiplications that either have no operand in common (doubling case) or share the operand Z_1 (addition case). Similarly to Chevallier-Mames *et al.*'s scheme, we can hence define the two following random variables:

$$\begin{cases} T_1 = \left(Z_1 \cdot Z_1\right)^{1-s} \cdot \left(Z_1 \cdot Z_1\right)^{s} \\ T_2 = \left(X_1 \cdot 4Y_1^2\right)^{1-s} \cdot \left(Z_1^2 \cdot Z_1\right)^{s} \end{cases}, \tag{4}$$

DOUBLING

$$R_1 \leftarrow X_1,\ R_2 \leftarrow Y_1,\ R_3 \leftarrow Z_1$$

$$1.\ \left[\begin{array}{l}
\boldsymbol{R_3 \leftarrow R_3^2} \quad (= Z_1 \cdot \boldsymbol{Z_1}) \\
\star \\
R_5 \leftarrow R_1 + R_4 \\
R_6 \leftarrow R_2^2 \\
R_4 \leftarrow -R_4 \\
R_2 \leftarrow R_2 + R_2 \\
R_4 \leftarrow R_1 + R_4
\end{array}\right.
\qquad
3.\ \left[\begin{array}{l}
R_5 \leftarrow R_4^2 \\
\star \\
R_6 \leftarrow R_2 + R_2 \\
\boldsymbol{R_6 \leftarrow R_1 \cdot R_6} \quad (= X_1 \cdot \boldsymbol{4Y_1^2}) \\
R_1 \leftarrow -R_6 \\
R_1 \leftarrow R_1 + R_1 \\
R_1 \leftarrow R_1 + R_5
\end{array}\right.$$

ADDITION
(mixed coordinates)

Input: $P = (X_1 : Y_1 : Z_1)$ and $Q = (X_2, Y_2)$
Output: $P + Q = (X_3 : Y_3 : Z_3 : X_1' : Y_1')$
$R_1 \leftarrow X_1,\ R_2 \leftarrow Y_1,\ R_3 \leftarrow Z_1,\ R_x \leftarrow X_2,\ R_y \leftarrow Y_2$

$$1.\ \left[\begin{array}{l}
\boldsymbol{R_4 \leftarrow R_3^2} \quad (= Z_1 \cdot \boldsymbol{Z_1}) \\
\star \\
\star \\
R_5 \leftarrow R_x \cdot R_4 \\
R_6 \leftarrow -R_1 \\
R_5 \leftarrow R_5 + R_6 \\
\star
\end{array}\right.
\qquad
3.\ \left[\begin{array}{l}
R_9 \leftarrow R_5 \cdot R_6 \\
\star \\
R_8 \leftarrow R_8 + R_9 \\
\boldsymbol{R_4 \leftarrow R_3 \cdot R_4} \quad (= Z_1^2 \cdot \boldsymbol{Z_1}) \\
\star \\
\star \\
\star
\end{array}\right.$$

Fig. 2. The first and third patterns used in atomicity of Longa

Under Assumption 1, it leads to the recovery of s.

Attack on Giraud and Verneuil's Scheme. Giraud and Verneuil introduced in [18] a new atomic pattern which reduces the number of field additions, negations and dummy operations ($\star$) compared to the above proposals. The patterns are recalled in Fig. 3.

Once again, depending on the secret s, we observe a repetition of two multiplications with a common operand in the first pattern of the addition scheme (ADD 1.), leading to the following equations:

$$\begin{cases} T_1 = \left(X_1 \cdot X_1\right)^{1-s} \cdot \left(Z_2 \cdot Z_2\right)^s \\ T_2 = \left(2Y_1 \cdot Y_1\right)^{1-s} \cdot \left(Z_2^2 \cdot Z_2\right)^s \end{cases}, \tag{5}$$

which, under Assumption 1, leads to the recovery of s.

Remark 3. A second version of the patterns in Fig. 3 has been proposed in [18] which allows to save more field additions and negations without addition of dummy operations. This proposal share the same weakness as the previous ones and our attack still applies.

ADDITION DOUBLING

$$\text{ADD 1.}\left[\begin{array}{l} R_1 \leftarrow Z_2 \cdot \boldsymbol{Z_2} \\ \star \\ \star \\ \star \\ R_2 \leftarrow X_1 \cdot R_1 \\ \star \\ \star \\ \star \\ R_1 \leftarrow R_1 \cdot \boldsymbol{Z_2} \quad (= Z_2^2 \cdot \boldsymbol{Z_2}) \\ \star \\ \star \\ \star \\ \vdots \end{array}\right. \qquad \text{ADD 2.}\left[\begin{array}{l} R_6 \leftarrow R_4^2 \\ \star \\ \star \\ \star \\ R_5 \leftarrow Z_1 \cdot Z_2 \\ \star \\ \star \\ \star \\ Z_3 \leftarrow R_5 \cdot R_4 \\ \star \\ \star \\ \star \\ \vdots \end{array}\right. \qquad \text{DOUB}\left[\begin{array}{l} R_1 \leftarrow X_1 \cdot \boldsymbol{X_1} \\ R_2 \leftarrow Y_1 + Y_1 \\ \star \\ \star \\ Z_2 \leftarrow R_2 \cdot Z_1 \\ R_4 \leftarrow R_1 + R_1 \\ \star \\ \star \\ R_3 \leftarrow R_2 \cdot Y_1 \quad (= 2Y_1 \cdot \mathbf{Y_1}) \\ R_6 \leftarrow R_3 + R_3 \\ \star \\ \star \\ \vdots \end{array}\right.$$

Fig. 3. The beginning of Giraud and Verneuil's patterns

Attack on Edward's Curves. Edward's representation of elliptic curves has been introduced in [17]. In a subsequent paper [6], Bernstein and Lange homogenized the curve equation in order to avoid field inversions in Edward's addition and doubling formulas. For this homogenized representation, points addition and doubling are both computed thanks to the same formula. Let $P = (X_1 : Y_1 : Z_1)$ and $Q = (X_2 : Y_2 : Z_2)$ be two points on the curve, the sum $R = (X_3 : Y_3 : Z_3)$ of P and Q is given by the following system:

$$\begin{cases} X_3 = Z_1Z_2(X_1Y_2 - Y_1X_2)(X_1Y_1Z_2^2 + Z_1^2X_2Y_2) \\ Y_3 = Z_1Z_2(X_1X_2 + Y_1Y_2)(X_1Y_1Z_2^2 - Z_1^2X_2Y_2) \\ Z_3 = dZ_1^2Z_2^2(X_1X_2 + Y_1Y_2)(X_1Y_2 - Y_1X_2) \end{cases},$$

where d is some constant related to the Edward curve equation. These formulae correspond to the sequence of operations given by Fig. 4.

This sequence also works when $P = Q$, meaning that it applies similarly for addition and doubling. This is one of the main advantage of Edward's representation compared to the other ones (e.g. Projectives) where such a unified formula does not exist. However it is significantly more costly than the separate addition and doubling formulas.[4]

Here, we can exploit the fact that the multiplication X_1Z_1 is performed twice if $P = Q$ (i.e. when the formula processed a doubling), which is not the case otherwise (see Fig. 4). We can hence define the two following random variables:

$$\begin{cases} T_1 = \left(X_1 \cdot Z_1\right)^{1-s} \cdot \left(X_1 \cdot Z_2\right)^s \\ T_2 = \left(X_1 \cdot Z_1\right)^{1-s} \cdot \left(X_2 \cdot Z_1\right)^s \end{cases}, \tag{6}$$

[4] Indeed, let us denote by M the cost of a field multiplication and by S the cost of a squaring. We assume $S = 0.8M$, which is usually satisfied in current implementations. For points in projective coordinates, the unified formulas for Weierstrass curves [10] require around $15.8M$ which represents a similar cost than for addition points (around $16M$) but is significantly higher than that of the doubling (around $9M$). The unified formula for Edward curves costs around $11M$ which is less than in the Weierstrass case but still higher than the classical formulas.

```
Addition
1. $R_1 \leftarrow \mathbf{X_1Z_2}$
2. $R_2 \leftarrow Y_1Z_2$
3. $R_3 \leftarrow \mathbf{Z_1X_2}$
4. $R_4 \leftarrow Z_1Y_2$
       $\vdots$
```

Fig. 4. First steps of algorithm for addition.

which, under Assumption 1, leads to the recovery of s.

Remark 4. This technique still applies in the case of other unified formulas (e.g. those introduced in [10]). Indeed, the sequence of operations in [10] present the same weaknesses as Edward's case. The multiplication X_1Z_1 is performed twice if the current operation is a doubling (see the first and third multiplications in [10, Sect. 3, Fig. 1]).

3.3 Distinguishing Common Operands in Multiplications

In this section we apply the collision attack principle presented in Sect. 3.1 to show how an adversary may deal with Assumption 1. This will conclude our attack description. As mentioned before, we assume that the field multiplications are implemented in an arithmetic co-processor with a *Long Integer Multiplication* (`LIM`) followed by a reduction. Many other multiplication methods exist but our attack can always be slightly adapted to also efficiently apply to those methods (see the full version of the paper).

Let ω denote an architecture size (e.g. ω equals 8, 16 or 32) and let us denote by $(X[t], \cdots, X[1])_{2^\omega}$ the base-2^ω representation of an integer. We recall hereafter the main steps of the `LIM` when applied between two integers X and Y.

Let W, X, Y and Z be four independent values of size $t\omega$ bits. We show hereafter how to distinguish by side-channel analysis the two following cases:

- Case (1) where the device processes `LIM`(X, W) and `LIM`(Y, Z) (all the operands are independent),
- Case (2) where `LIM`(X, Z) and `LIM`(Y, Z) are processed (the two `LIM` processings share an operand).

For such a purpose, and by analogy with our side-channel model in Sect. 2.1 and Table 1, we denote by C_1 (resp. C_2) the multiplication in the loop during the first `LIM` processing (resp. the second `LIM` processing) and by O_1 (resp. O_2) its result. The output of each multiplication during the loop may be viewed as a realization of the random variable O_1 (resp. O_2). To each of those realizations we associate a leakage $\ell^1_{a,b}$ (resp. $\ell^2_{a,b}$). To distinguish between cases (1) and (2), we directly apply the attack described in Table 1 and we compute the Pearson's correlation coefficient:

$$\hat{\rho}\Big((\ell^1_{a,b})_{a,b}, (\ell^2_{a,b})_{a,b}\Big) \ . \tag{7}$$

Algorithm 1. Long Integer Multiplication (LIM)

```
Input: $X = (X[t], X[t-1], \ldots, X[1])_{2^\omega}$, $Y = (Y[t], Y[t-1], \ldots, Y[1])_{2^\omega}$.
Output: $\mathtt{LIM}(X, Y)$.
for $a$ from 1 to $2t$ do
    $R[a] \leftarrow 0$
for $a$ from 1 to $t$ do
    $C \leftarrow 0$
    for $b$ from 1 to $t$ do
        $(U, V)_{2^\omega} \leftarrow X[a] \cdot Y[b]$          // Operation $\mathsf{C}_1$ (resp. $\mathsf{C}_2$)
        $(U, V)_{2^\omega} \leftarrow (U, V)_{2^\omega} + C$
        $(U, V)_{2^\omega} \leftarrow (U, V)_{2^\omega} + R[a+b-1]$
        $R[a+b-1] \leftarrow V$
        $C \leftarrow U$
    $R[a+t] \leftarrow C$
return $R$
```

In place of (7), the following correlation coefficient can be used in the attack:

$$\hat{\rho}\left(\left(\frac{1}{t}\sum_a \ell^1_{a,b}\right)_b, \left(\frac{1}{t}\sum_a \ell^2_{a,b}\right)_b\right) . \tag{8}$$

In the following section we actually argue that this second correlation coefficient gives better results, which is confirmed by our attacks simulations reported in Sect. 4.

3.4 Study of the Attack Soundness

This section aims at arguing on the soundness of the approach described previously to distinguish common operands in multiplications. For such a purpose, we explicit formulae for the correlation coefficients given in (7) and (8). For simplicity, the development is made under the assumption that the device leaks the Hamming weight of the processed data but similar developments could be done for other models and would lead to other expressions. Under the Hamming weight assumption, we have $\ell^1_{a,b} \hookleftarrow \mathrm{HW}(O_1) + B_1$ and $\ell^2_{a,b} \hookleftarrow \mathrm{HW}(O_2) + B_2$ where B_1 and B_2 are two independent Gaussian random variables with zero mean and standard deviation σ.

- If O_1 and O_2 correspond to the internal multiplications during the processings of $\mathtt{LIM}(X,\ W)$ and $\mathtt{LIM}(Y,\ Z)$ respectively, then, for every $(a, b) \in [1; t]^2$, we have:

$$\ell^1_{a,b} = \mathrm{HW}(x[a] \cdot w[b]) + b_{1,a,b} \tag{9}$$

$$\ell^2_{a,b} = \mathrm{HW}(y[a] \cdot z[b]) + b_{2,a,b} \ . \tag{10}$$

  Since W, X, Y and Z are independent, the correlation coefficients in (7) and (8) tend towards 0 when t tends towards infinity.

- If O_1 and O_2 correspond to the internal multiplications during the processings of $\mathtt{LIM}(X,\, Z)$ and $\mathtt{LIM}(Y,\, Z)$ respectively, then we have:

$$\ell^1_{a,b} = \mathrm{HW}(x[a] \cdot z[b]) + b_{1,a,b} \tag{11}$$
$$\ell^2_{a,b} = \mathrm{HW}(y[a] \cdot z[b]) + b_{2,a,b} \ . \tag{12}$$

  Since the two multiplications share an operand, their results are dependent. In this case indeed, it can be proved that the correlation coefficients (7) and (8) satisfy:

$$\hat{\rho}\Big((\ell^1_{a,b})_{a,b}, (\ell^2_{a,b})_{a,b}\Big) \simeq \frac{1}{1+\frac{2^{2\omega+2}\sigma^2+(\omega-1)2^{2\omega}+2^{\omega}}{2.2^{2\omega}-(2\omega+1)2^{\omega}-1}}$$

  and

$$\hat{\rho}\left(\Big(\frac{1}{t}\sum_a \ell^1_{a,b}\Big)_b, \Big(\frac{1}{t}\sum_a \ell^2_{a,b}\Big)_b\right) \simeq \frac{1}{1+\frac{1}{t}\frac{2^{2\omega+2}\sigma^2+(\omega-1)2^{2\omega}+2^{\omega}}{2.2^{2\omega}-(2\omega+1)2^{\omega}-1}} .$$

  When t tends towards infinity, it may be noticed that the second correlation coefficient tends towards 1 (which is optimal).

4 Experiments

In order to validate the approach presented in Sect. 3.3 and thus to illustrate the practical feasibility of our attack, we performed several simulation campaigns for various sizes of elliptic curves, namely $\lceil \log_2(p) \rceil \in \{160, 256, 384\}$, implemented on different kinds of architectures, namely $\omega \in \{8, 32\}$ using the Chevallier-Mames *et al.*'s scheme. Each experiment has been performed in the same way. For each (p, ω), we computed Pearson's correlation coefficients (7) and (8) between the sample of observations coming from the leakages on operations C_1 and C_2 in the two following cases:

- when the secret bit s is equal to 1, that is when an addition is performed (which implies correlated random variables, see (3)),
- when the secret bit s is equal to 0, that is when a doubling operation is performed (which implies independent random variables, see (3)).

From the configuration (p, ω), the size t of the observations' samples used in the attack can be directly deduced: it equals $\lceil \frac{\log_2(p)}{\omega} \rceil$. The quality of the estimations of the correlation coefficient by Pearson's coefficient depends on both the observations *signal to noise ratio* (SNR) and t. When the SNR tends towards 0, the sample size t must tend towards infinity to deal with the noise. Since, in our attack the samples size cannot be increased (it indeed only depends on the implementation parameters p and ω), our correlation estimations tend towards zero when the SNR decreases. As a consequence, distinguishing the two Pearson coefficients coming from $s = 0$ and $s = 1$ becomes harder when the SNR decreases.

This observation raises the need for a powerful (and robust to noise) test to distinguish the two coefficients. To take this into account for each setting (p, ω) and several SNR, we computed the mean and the variance of Pearson's coefficient defined in (7) and (8) over 1000 different samples of size t. To build those kinds of templates, leakages have been generated in the Hamming weight model with additive Gaussian noise of mean 0 and standard deviation σ (i.e. according to (9)-(10) for $s = 0$ and to (11)-(12) for $s = 1$)[5]. When there is no noise at all, namely when $\sigma = 0$ (i.e. $\texttt{SNR} = +\infty$), one can observe that the mean of Pearson's coefficient is coherent with the predictions evaluated in Sect. 3.4.

Figures (5, 6, 7, 8) illustrate the spreading of the obtained Pearson's coefficient around the mean value. This variance gives us information about the amount of trust we can put into the mean values. It also shows whether a distinction between the right hypothesis and the wrong one can easily be highlighted. For each SNR value (denoted by τ) and each sample size t, let us denote by $\hat{\rho}_{0,t}(\tau)$ (resp. $\hat{\rho}_{1,t}(\tau)$) the random variable associated to the processing of (7) for $s = 0$ (resp. for $s = 1$). In Figs. (5, 6, 7, 8), we plot estimations of the mean and variance of $\hat{\rho}_{0,t}(\tau)$ and $\hat{\rho}_{1,t}(\tau)$ for several pairs (τ, t). Clearly, the efficiency of the attack described in Sect. 3 depends on the ability of the adversary to distinguish, for a fixed pair (t, τ), the distribution of $\hat{\rho}_{0,t}(\tau)$ from that of $\hat{\rho}_{1,t}(\tau)$. In other terms, once the adversary has computed a Pearson coefficient $\hat{\rho}$ he must decide between the two following hypotheses; $\mathrm{H}_0 : \hat{\rho} \hookleftarrow \hat{\rho}_{0,t}(\tau)$ or $\mathrm{H}_1 : \hat{\rho} \hookleftarrow \hat{\rho}_{1,t}(\tau)$. For such a purpose, we propose here to apply a *maximum likelihood* strategy and to choose the hypothesis having the highest probability to occur. This led us to approximate the distribution of the coefficients $\hat{\rho}_{0,t}(\tau)$ and $\hat{\rho}_{1,t}(\tau)$ by a Gaussian distribution with mean and variance estimated in the Hamming weight model (as given in Figs. 5, 6, 7, 8). Attacks reported in Figs. 9 and 10 are done with this strategy.

Remark 5. Since the adversary is not assumed to know the exact leakage SNR, the maximum likelihood can be computed for several SNR values τ starting from ∞ to some pre-defined threshold. This problematic occurs each time that the principle of collision attacks is applied.

Remark 6. For a curve of size $n = \lceil \log_2(p) \rceil$ and a ω-bit architecture, the adversary can have a sample of $t = \lceil \frac{n}{\omega} \rceil$ observations if he averages over the columns and $t = \lceil (\frac{n}{\omega})^2 \rceil$ without averaging. All experiments provided in this section have been performed using the "average" strategy.

This attack works for any kind of architecture, even for a 32-bit one (see Fig. 10), which is the most common case in nowadays implementations. In the presence of noise, the attack success decreases highly but stays quite successful for curves of size 160, 256 and 384 bits. In all experiments (Figs. 9, 10), we also observe that the success rate of our attack increases when the size of the curve becomes larger. This behaviour can be explained by the increasing number of observations available in this case. Paradoxically, it means that when the

[5] In this context, the SNR simply equals $\omega/4\sigma^2$.

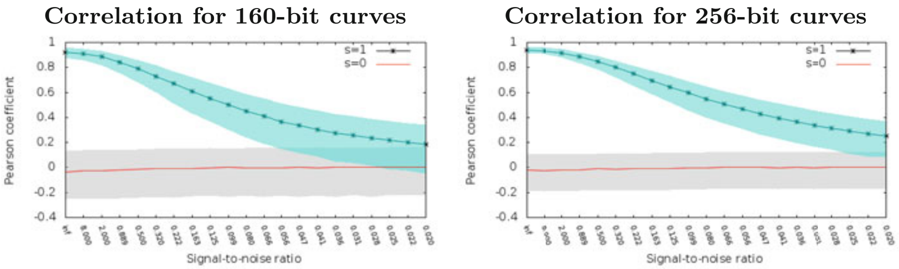

Fig. 5. Pre-computations on $w = 8$-bit registers

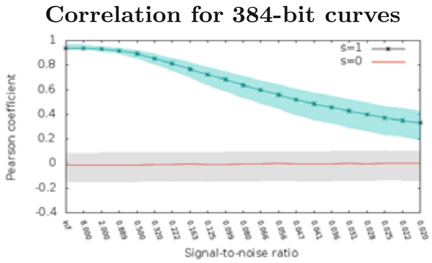

Fig. 6. Pre-computations on $w = 8$-bit registers

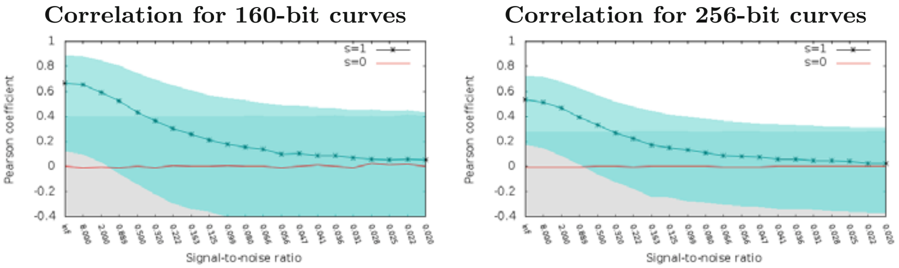

Fig. 7. Pre-computations on $w = 32$-bit registers

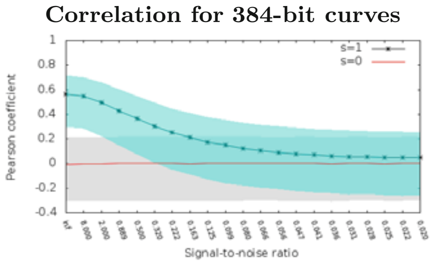

Fig. 8. Pre-computations on $w = 32$-bit registers

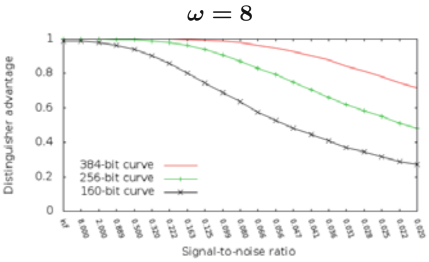

Fig. 9. Success rate of the attack on 8-bit registers

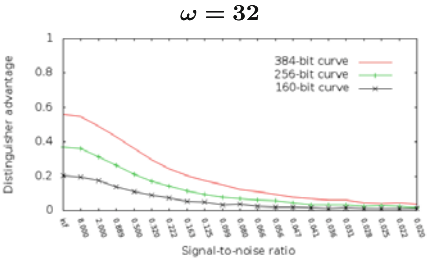

Fig. 10. Success rate of the attack on 32-bit registers

theoretical level of security becomes stronger (i.e. p is large), resistance against side-channel attacks becomes weaker. This fact stands in general for horizontal attacks and has already been noticed in [12,33].

References

1. ANSI X9.62: Public Key Cryptography for The Financial Service Industry : The Elliptic Curve Digital Signature Algorithm (ECDSA). American National Standards Institute (1998)
2. ANSI X9.63: Public Key Cryptography for The Financial Service Industry : Key Agreement and Key Transport Using Elliptic Curve Cryptography. American National Standards Institute (1998)
3. Batina, L., Gierlichs, B., Prouff, E., Rivain, M., Standaert, F.-X., Veyrat-Charvillon, N.: Mutual information analysis: a comprehensive study. J. Cryptol. **24**(2), 269–291 (2011). (to appear)
4. Bauer, A., Jaulmes, E., Prouff, E., Wild, J.: Horizontal and vertical side-channel attacks against secure RSA implementations. In: Dawson, E. (ed.) CT-RSA 2013. LNCS, vol. 7779, pp. 1–17. Springer, Heidelberg (2013)
5. Bernstein, D.J., Lange, T.: Analysis and optimization of elliptic-curve single-scalar multiplication. Cryptology ePrint Archive, Report 2007/455 http://eprint.iacr.org/ (2007)
6. Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 29–50. Springer, Heidelberg (2007)

7. Billet, O., Joye, M.: The Jacobi model of an elliptic curve and side-channel analysis. Cryptology ePrint Archive, Report 2002/125 (2002)
8. Bogdanov, A., Kizhvatov, I., Pyshkin, A.: Algebraic methods in side-channel collision attacks and practical collision detection. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 251–265. Springer, Heidelberg (2008)
9. Brier, E., Clavier, Ch., Olivier, F.: Correlation power analysis with a leakage model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
10. Brier, E., Joye, M.: Weierstraß elliptic curves and side-channel attacks. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 335–345. Springer, Heidelberg (2002)
11. Chevallier-Mames, B., Ciet, M., Joye, M.: Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity. IEEE Trans. Comput. **53**(6), 760–768 (2004)
12. Clavier, Ch., Feix, B., Gagnerot, G., Giraud, Ch., Roussellet, M., Verneuil, V.: ROSETTA for single trace analysis. In: Galbraith, S., Nandi, M. (eds.) INDOCRYPT 2012. LNCS, vol. 7668, pp. 140–155. Springer, Heidelberg (2012)
13. Clavier, Ch., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Horizontal correlation analysis on exponentiation. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 46–61. Springer, Heidelberg (2010)
14. Clavier, Ch., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Improved collision-correlation power analysis on first order protected AES. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 49–62. Springer, Heidelberg (2011)
15. Cohen, H., Frey, G. (eds.): Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, Baco Raton (2005)
16. Coron, J.-S.: Resistance against differential power analysis for elliptic curve cryptosystems. In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 292–302. Springer, Heidelberg (1999)
17. Edwards, H.M.: A normal form for elliptic curves. Bull. Am. Math. Soc. **44**, 393–422 (2007)
18. Giraud, Ch., Verneuil, V.: Atomicity improvement for elliptic curve scalar multiplication. In: Gollmann, D., Lanet, J.-L., Iguchi-Cartigny, J. (eds.) CARDIS 2010. LNCS, vol. 6035, pp. 80–101. Springer, Heidelberg (2010)
19. Hankerson, D., Menezes, A.J., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer Professional Computing Series. Springer, New York (2003)
20. ISO/IEC JTC1 SC17 WG3/TF5 for the International Civil Aviation Organization. Supplemental Access Control for Machine Readable Travel Documents. Technical Report (2010)
21. Koblitz, N.: Elliptic curve cryptosystems. Math. Comput. **48**(177), 203–209 (1987)
22. Koç, Ç.K., Naccache, D., Paar, C. (eds.): CHES 2001. LNCS, vol. 2162. Springer, Heidelberg (2001)
23. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)
24. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
25. Liardet, P.-Y., Smart, N.P.: Preventing SPA/DPA in ECC systems using the Jacobi form. In: Koç, Ç.K., et al. (eds.) [22], pp. 401–411

26. Longa, P.: Accelerating the scalar multiplication on elliptic curve cryptosystems over prime fields. Master's thesis, School of Information Technology and Engineering, University of Ottawa, Canada (2007)
27. Messerges, T.S.: Using second-order power analysis to attack DPA resistant software. In: Paar, Ch., Koç, Ç.K. (eds.) CHES 2000. LNCS, vol. 1965, pp. 238–251. Springer, Heidelberg (2000)
28. Micali, S., Reyzin, L.: Physically observable cryptography. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 278–296. Springer, Heidelberg (2004)
29. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)
30. Moradi, A.: Statistical tools flavor side-channel collision attacks. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 428–445. Springer, Heidelberg (2012)
31. Moradi, A., Mischke, O., Eisenbarth, T.: Correlation-enhanced power analysis collision attack. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 125–139. Springer, Heidelberg (2010)
32. Schramm, K., Wollinger, T., Paar, Ch.: A new class of collision attacks and its application to des. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 206–222. Springer, Heidelberg (2003)
33. Walter, C.D.: Sliding windows succumbs to big mac attack. In: Koç, Ç.K., et al. (eds.) [22], pp. 286–299