Survey for Performance & Security Problems of Passive Side-channel Attacks Countermeasures in ECC

Rodrigo Abarúa[1], Claudio Valencia[2], and Julio López[3]

[1]Departamento de Matemáticas y Ciencia de la Computación, Universidad de Santiago de Chile , rodrigo.abarzua@usach.cl
[2] Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile.
claudio.valenciac@usach.cl
[2]Julio López, Instituto de Computaçao, Universidade Estadual de Campinas,
jlopez@ic.unicamp.br

January 4, 2019

Abstract

The main objective of the Internet of Things is to interconnect everything around us to obtain information which was unavailable to us before, thus enabling us to make better decisions. This interconnection of things involves security issues for any Internet of Things key technology. Here we focus on elliptic curve cryptography (ECC) for embedded devices, which offers a high degree of security, compared to other encryption mechanisms. However, ECC also has security issues, such as Side-Channel Attacks (SCA), which are a growing threat in the implementation of cryptographic devices. This paper analyze the state-of-the-art of several proposals of algorithmic countermeasures to prevent passive SCA on ECC defined over prime fields. This work evaluates the trade-offs between security and the performance of side-channel attack countermeasures for scalar multiplication algorithms without pre-computation, i.e. for variable base point.

Although a number of results are required to study the state-of-the-art of side-channel attack in elliptic curve cryptosystems, the interest of this work is to present explicit solutions that may be used for the future implementation of security mechanisms suitable for embedded devices applied to Internet of Things. In addition security problems for the countermeasures are also analyzed.

Keywords: Internet of Things, Elliptic curve cryptosystems, Side-channel attack, Countermeasures.

1 Introduction

At present, it is not unusual to have our mobile and home devices connected through the development of different devices and communication protocols such as the wireless sensor networks (WSN), Radio - Frequency Identification (RFID) and miniaturization technologies, which enable generating new technology embedded and immersed in our daily lives, known as Internet of

Things (IoT). The application ranges of IoT are completely intersected, ranging from industrial automation to remote care of people [116].

The features of IoT and the broad variety of devices for each application enable opening lines of research to cover different areas of knowledge, including security, which represents a fundamental problem in our days [68, 117]. A central component in the embedded IoT are microcontrollers, therefore incorporating security mechanisms for these components is relevant due to their characteristics.

Public-key cryptography (PKC) plays an important role in embedded IoT devices to provide security services such as confidentiality, data authentication and key exchange [137]. A clear example of PKC is the Elliptic Curve Cryptography (ECC) proposed by Koblitz [77] and Miller [89], which provides the same cryptographic strength as the RSA public-key system with significant smaller key sizes. For example, a 256-bit ECC key is equivalent to RSA 3072-bit key. Due to smaller key sizes, ECC offers some advamtages for compact and faster implementation on embedded devices [50]. Therefore, the ECC is a viable option for IoT applications. In general, the development of techniques for protecting cryptographic algoritms against Side Channel Attack (SCA) is crucial for the security of applications running on microcontrollers used in IoT.

Passive SCA exploit physical leakages on a device when a cryptographic process is executed, where we have timing [80], power consumption [78] and electromagnetic radiation [107, 48] attacks. These attacks are easy to perform in microcontrollers without proper countermeasures.

There are two general strategies to these attacks: *Simple Side-channel Analysis* (SSCA) [80] and *Differential Side-channel Analysis* (DSCA) [78]. The SSCA analyses the differences of physical leakages on a device using a single scalar multiplication, otherwise the DSCA uses statistical techniques to retrieve information based on the measurements of physical leakages on a device using several scalar multiplications [104].

There are currently several studies on SCA and countermeasures in ECC such as the Avanzi's report [6], book [29], surveys [41, 39, 34] and PhD theses [36, 38, 132, 133]. It is noticed that previous works, in general, present studies from a point of view of the attacks and countermeasures, and not the security analysis of the countermeasures.

This article is an extension version of the work presented in [98]. The objective is to show a panorama of solutions that designers and implementors can choose to protect ECC against SCA, targeted a very restricted embedded devices suitable for IoT. Our focus is in protecting the central operation of ECC, which is: given an non-negative integer k and an elliptic point P, compute $[k]P$. This operation is known as point multiplication or scalar multiplication. The main contribution is based on our security analysis of the SCA countermeasures on ECC, providing a comparative analysis between security versus computational cost.

The characteristics of embedded IoT devices limit the paper scope to SCA countermeasures that do not use precomputation tables to store elliptic curve points.

The paper is organized as follows. In the next Section, we present a brief background of elliptic curves. Scalar multiplication algorithms are presented in Section 3 and the countermeasure of different passive attacks are given in Sections 4, 5, 6, 7, 8 and 9. Futhermore a summary of the countermeasures and security problems are presented in Section 10. Finally, we present the conclusion in Section 11.

2 Mathematical Background

An elliptic curve E defined over a large prime field $\mathbb{F}_p$ is given by an equation of the form $E(\mathbb{F}_p) : y^2 = x^3 + ax + b$, with $4a^3 + 27b^2 \neq 0$. The group used for cryptography consists of the affine ($\mathcal{A}$) points (x, y) on the curve and the point at infinity P_∞ (the neutral element), with the "chord-and-tangent" addition. The group operation for $(x_1, y_1) + (x_2, y_2)$ is given by:

$$(x_3, y_3) = \left(\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1\right),$$

where,

$$\lambda = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2} & \text{if } x_1 \neq x_2, \quad [\mathbf{ECADD}^{\mathcal{A}}] \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } (x_1, y_1) = (x_2, y_2). \quad [\mathbf{ECDBL}^{\mathcal{A}}] \end{cases}$$

and $(x_1, y_1) + (x_1, -y_1) = P_\infty$.

The (projective) points on E_1 are divided into classes by the equivalence relation: $(X_1, Y_1, Z_1) \equiv (X_2, Y_2, Z_2) \iff \exists \lambda \in \mathbb{F}_p^*$ s.t. $(X_2, Y_2, Z_2) = (\lambda^2 X_1, \lambda^3 Y_1, \lambda Z_1)$. We denote by $(X : Y : Z)$ the equivalence class containing the point (X, Y, Z). Note that every equivalence class with $Z \neq 0$ contains exactly one point of the form $(x, y, 1)$ which corresponds to an point (x, y) on curve E (and vice versa). The (unique) equivalence class with $Z = 0$ is of the form $(\rho^3 : \rho^2 : 0)$ (with $\rho \in \mathbb{F}_p^*$) and corresponds to the point at infinity P_∞ of E. A detail description of ECC is presented in [29, 56, 136].

The point addition formula is based on different operations over $\mathbb{F}_q$: multiplication (M), squaring (S), inversion (I), addition and subtraction, which have different computational costs. For a typical software implementation of prime field operations, it is often assumed that $I \approx 100M$ and $S = 0.8M$ [47]. In general, for a microcontroller, a field inversion costs a few dozens of field multiplications, the cost of a field squaring is slightly lower than the cost of a field multiplication and the cost of an addition/subtraction is significantly lower than a multiplication.

3 Scalar Multiplications Algorithm

The fundamental algorithm for ECC is the scalar multiplication $[k]P$, where k is an integer and P an elliptic curve point. In this paper, we concentrate mostly on the basic double-and-add method since it can be implemented without lookup tables. There are two versions of the basic double-and-add algorithm the *Right-to-left* and *Left-to-right*, in the Algorithm 1 the Left-to-right is showed.

Given the binary representation of an integer $k = \sum_{i=0}^{n-1} k_i 2^i$, the scalar multiplication $[k]P = (k_{n-1}2^{k-1} + \cdots + k_0 2^0)P$ is computed using the Horner's rule, therefore $[k]P = [k_0 + 2(k_1 + 2(\ldots(k_{n-2} + 2k_{n-1})\ldots))]P$ require n doublings and $n/2$ additions on average, denoted by $(n/2)A + nD$. The latter is true for the Algorithm 1 and *Right-to-left* Algorithm. Then the double-and-add method is optimal [29, 56, 25, 81].

The *Left-to-right* and *Right-to-left* binary methods may be subject to SCA. An adversary can to distinguish from a power trace between point doubling ($2R_i$, with $i = 0, 1$) and point addition ($R_0 + R_1$), and so it can recover the value of the secret scalar $[k]$ (for details see [80]).

The attack is possible because the algorithms used to compute $[k]P$ directly depends on the bits of the secret key k. For example, a general description of a SSCA can be explained

Algorithm 1 Double-and-add binary expansion method: Left-to-right

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Outputs: $\mathbf{Q} = [k] \cdot \mathbf{P}$
1: $R_0 \leftarrow P_\infty$; $R_1 \leftarrow P$
2: **for** i **from** $n-1$ **to** 0 **do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad$ **if** $k_i = 1$ **then**
5: $\quad\quad R_0 \leftarrow R_0 + R_1$
6: $\quad$ **end if**
7: **end for**
8: **return** R_0

with the following idea: if the bit of the secret key k is 1, both algorithms – *Left-to-right* and *Right-to-left-* compute one doubling and one addition. Otherwise, if the bit of the secret key k is 0, both algorithms only perform a doubling $2P$. These two operations have different curves of: time, power consumption or electromagnetic radiation. Then, observing these curves (power consumption, for example), we can found if the bit (in a given time) of the key k is 1 or 0, and then an attacker could be able to determine the bits of the secret key. Alternatively, DSCA [78] uses statistical techniques to retrieve information on the secret key based on measurements from several scalar multiplications [104]. For details see [6, 41, 70]. The following section the countermeasures of SSCA are will present.

4 Countermeasures of Simple Side-channel attacks in ECC

SSCA is made easier for $[k]P$ algorithms because the operations ECADD and ECDBL are different. Countermeasures to special elliptic curves are known such as *Edwards curves* [37, 12, 13, 61], *inverted Edwards curves* [14], *Huff model* curve [72], *Hessian curves* [113, 107, 42], *Jacobi curves* [84, 16, 35, 55, 62]. Recently, the Edwards curves were standardized [83]. This special family of elliptic curves is not studied in this work. Although, in general, we can choose an EC of the special form, it is very likely for us to select EC recommended by a standard. For example, over a large prime field, the NIST [101] and SEC 2 [114] recommends the use of prime order EC.

The usual way to prevent SSCA consists always repeating the same pattern of operation, whatever the point is processed. For example, the *double-and-add-always* algorithm of Coron [24] ensuring that the operation of the secret scalar are independent by inserting dummy ECADD between consecutive ECDBL. Other c ountermeasures to prevent SSCA are the following: *Unified Formulæ* of Brier-Joye in [18] and Brier-Dechene-Joye in [11], *Montgomery Ladder* over prime fields [18, 94, 63, 44] and fields of characteristic two [87], *Joye's Double-add, Add-Only* [73], *Zero-less Signed-digit expansion* (ZSD) in [52], *Atomic Blocks* [47, 20, 82, 26, 1]. Another approach consists in using "regular" representations of the scalar [95, 126, 73], with the same fixed pattern of group operations for all scalars. Note, that this family of countermeasures is not analyzed in this work.

In the next subsections, we present a security analysis and theoretical computation cost of the above mentioned countermeasures.

4.1 Unified Formulæ of Brier-Joye [18]

An *Unified formula* uses the same set of field operations for ECADD and ECDBL. For Weierstrass elliptic curves, we have the following algorithm for point addition [18] (for more details see [15]). The computational cost for an addition is $13M + 5S$.

Let $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$, with $x_i = X_i/Z_i$ and $y_i = Y_i/Z_i$, $R = P + Q = (X_3, Y_3, Z_3)$ is:

$$\begin{array}{llllll} U_1 = X_1Z_2, & U_2 = X_2Z_1, & S_1 = Y_1Z_2, & S_2 = Y_2Z_1, & & T = U_1 + U_2, \\ M = S_1 + S_2, & Z = Z_1Z_2, & F = ZM, & L = MF, & & G = TL, \\ R = T^2 - U_1U_2 + aZ^2, & W = R^2 - G, & X_3 = 2FW, & Y_3 = R(G - 2W) - L^2, & & Z_3 = 2F^3. \end{array}$$

Security Problems:

4.1.1 Izu-Takagi Attack

The *Unified Formulas* of Brier-Joye are only valid if $y_1 + y_2 \neq 0$. Izu and Takagi [66] presented an attack using two points such that $x_1 \neq x_2$ and $y_1 + y_2 = 0$. The main idea of the attack is to use an special point, which causes a fault, i.e. a division by zero ($0^{-1} \notin \mathbb{F}_p$) in converting from projective to affine coordinates at the end of the $[k]P$. The secret scalar k is thus guessed from the error of $[k]P$ for different points P. If an attacker wants to know $[m]P + P$ with $2 \leq m < k$, he can use a point P in such a way that $y(mP) + y(P) = 0$ (*the m-th self-collision point*). If the device replies with an fault to the attacker, he then knows the device compute $[m]P + P$. Starting with $m = 2$ [1], and by following this process, the attacker is able to recover the k bit-by-bit, from the most to the least significant.

4.1.2 Walter's Attack

The Walter's attack in [134] uses the side-channel leakage of the conditional substraction in a Montgomery modular multiplication (MMM) operation. Let $P = (X, Y, Z)$ a point in EC, when the doubling algorithm of Brier-Joye is calculated the registers $U_1 = U_2 = XZ$ are identical, and they exhibit identical side-channel leakage of the conditional substraction in MMM operation (the same property holds for S_1 and S_2). The behavior for point addition is different. When the addition algorithm is used, the input point $P = (X_1, Y_1, Z_1)$ are randomized coordinates, the latter implies that occasionally Z_1 will be large and X_1 and Y_1 will both be small. Then, the computations of U_1 and S_1 are less likely to include the additional subtraction in MMM, on the other hand, the computations of U_2 and S_2 are more likely to include the additional subtraction. This difference in behavior can be detected by a side-channel attack and used accordingly to recover the bits of k.

Recently Wang *et al.* [135] performs an implementation of Walter's attack on a smart card verifying that Walter's attack is effective against *Double-and-Add Always* and the *Montgomery Ladder* Algorithm.

4.1.3 Amiel *et al*'s Attack

A common requirement for several countermeasures against SSCA is that M and S fields operations are indistinguishable from the SCA point of view. Particularly, the countermeasures

[1] When $m = 2$ and the attacker knows whether $y(2P) + y(P) = 0$, then, if it is, $k_{n-2} = 1$; otherwise, $k_{n-2} = 0$

atomic blocks [47, 20, 26] and *Unified Formulas* [12, 72, 18] assumes this property. However, this assumption is not always true. Amiel's attack [4] is based on distinguishing between M and S using the power consumption trace. This is possible because the Hamming weight of the result of a M and S are different, and they can be distinguished in the power traces.

Notice that when a point addition is computed, the multiplications $Z = Z_1Z_2$ and U_1U_2 are different but when a point doubling operation is computed the above operations are $Z = Z_1^2$ and U_1^2, hence, Amiel's attack can be applied to such implementations.

4.1.4 Combined Attack, Passive and Active Attack (PACA)

Amiel *et al.* in [5] presents a combined attack on a resistant implementation to side channel of RSA. This attack is easily applied to ECC.

The PACA attack use the following idea. An attacker applies a fault in the register that store the Z coordinate of point P_1, say $Z_1 = 0$, then, we use the *Unified Formula* which has two different patterns for the calculation, $Z = Z_1 \cdot Z_1 = 0 \cdot 0$ (if doubling is computed) and $Z = Z_1 \cdot Z_2 = 0 \cdot Z_2$ (with $Z_2 \neq 0$ if addition is computed). The authors report that these two patterns can be identified in a power consumption trace [5, 118]. The latter allows an attacker applying SPA techniques to distinguish between additions operations and doubling operations to know the secret key k. Schmidt *et al.* in [118] present this attack in Edwards curves [37] in inverted coordinates, and this attack is to apply *Unified Formula* of Brier-Joye.

4.1.5 Horizontal Collision correlation analysis (HCCA)

Recently, Bauer *et al.* [9] presents the HCCA, this new attack use two different techniques called *Horizontal power analysis* and *Collision correlation analysis* which are effective for *Atomic Blocks* and *Unified formulae* countermeasures. The main assumption is that *"The adversary can detect when two field multiplication have at least one operand in common"*, basically, when the *Unified Formula* performs a doubling operation ($P = Q$) the attack uses the fact that the multiplication X_1Z_2 is computed twice (U_1 and U_2), hence, we can define the two following applications: $T_1 := (X_1 \cdot Z_1)^{1-s} \cdot (X_1 \cdot Z_2)^s$, $\quad T_2 := (X_1 \cdot Z_1)^{1-s} \cdot (X_2 \cdot Z_1)^s$. In their work, they assume a co-processor with *Long Integer Multiplication* (LIM) implemented, followed by field reduction. Using the Hamming weight leakage model in order to demonstrate from theoretical and practical points of view that there is a Pearson's linear correlation (Pc) of LIM(X, Z) and LIM(Y, Z) (i.e. two LIM processing share the same operand), when $s = 0 \rightarrow Pc \approx 1$ then there is correlation. When the device processes LIM(X, W) and LIM(Y, Z) (all the operands are independent), hence $s = 1 \rightarrow Pc \approx 0$ therefore there is no correlation.

4.1.6 Horizontal Same Value Attack (SVA)

Murdica in [97] presents SVA for ECC applying vertical attack technique. Later, Danger in [32] use the SVA for ECC apply horizontal attack technique, if the input point of *Unified Formula* are the same $P = Q$, the values U_1 and U_2 are equal, the same occurs for the S_1 and S_2 values. Therefore, this attack can differentiate a doubling or addition and only requires a single trace, for example energy, to reveal all the secret bits of scalar k. For more details see [32].

Observation Table 11 shows a relationship between the types of attacks to *Unified Formulas* versus the requirements to perform the attack. Here, we can see that Izu-Takagi attack [66]

needs a larger set of requirements to achieve the attack. On the other hand, the remaining attacks need a simple execution to perform the attack. It is noted that the required statistical analysis cost to perform the attack has not been considered.

Recently, Renes *et al.* in [108] presented new complete formulas for elliptic curves on prime field. The new formulas naturally protect against SSCA, the computational cost for curves of type $a = -3$ used in standards NIST and SEGC is $12M + 2m_b + 29A$ where m_b is defined as multiplication by the b parameter of the elliptic curve. Chmielewski *et al.* in [30] presents an implementation of the Renes formulas in an FPGA platform, and they show in a real way that these formulas protect against SSCA.

Futhermore, Das in [33] shows that the complete formulas recently presented by Renes are resistant to Bauer attack [9], but that formula do not resist to the triangular analysis attacks [109] which exploits the inner collisions within a LIM, this attack is able to differentiate M or S field operations. Additionaly, Das presents the countermeasures which compute the square S and exchange the operands in a LIM and that can hide side channel information, for details of the algorithm see [33].

Moreover, Brier, Dechene and Joye in [11] presents a new *Unified Formula* to protect against the Izu-Takagi attacks. The cost of these new formulae is $16M + 3S$. Stebila and Thériault in [115] generalize Walter's attack and detect a conditional addition at the end of the Montgomery multiplication. Moreover, the Brier-Dechene-Joye formula is prone to Amiel's attacks [4] and PACA [5].

4.2 Double-and-add Always Coron's Algorithm

The *double-and-add-always* algorithm presented in [24] (Algorithm 2) uses an dummy point addition when the scalar bit $k_i = 0$ and the sequence of operations to compute a $[k]P$ is independent from the value of k. Thus, an adversary cannot guess the information bit of k_i by the SPA. A

Algorithm 2 *Double-and-add always* resistant against SPA

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Outputs: $Q = [k] \cdot P$
1: $R_0 \leftarrow P_\infty$
2: **for** i **from** $n-1$ **to** 0 **do do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad R_1 \leftarrow R_0 + P$
5: $\quad R_0 \leftarrow R_{k_i}$
6: **end for**
7: return R_0

drawback of this method is its efficiency, the algorithm requires $nA+nD$. The method increases the amount of field operations by the "dummy" computation in 33%.

Security Problems:

4.2.1 Fouque et.al. Doubling Attacks

The doubling attack by Fouque et.al. in [40] is an attack with chosen inputs, based on the fact that the similar intermediate values appear when the $[k]P$ is computed for input P and $2P$. Therefore, with two microcontroller requests, all the bits of the secret scalar may be recovered. The doubling attack is a powerful attack for some classic SPA-protected algorithms,

such as *double-and-add-always algorithm* including those using the blinding countermeasures [24]. The Algorithm 2 the partial sums are computed as follow, in the iteration j we get: $S_j(P) = \sum_{i=0}^{j} k_{n-i}2^{j-i}P = \sum_{i=0}^{j-1} k_{n-i}2^{j-1-i}(2P) + k_{n-j}P = S_{j-1}(2P) + k_{n-j}P$. Thus, the intermediate result of the Algorithm 2 with P at step j will be equal to the intermediate result with $2P$ at step $j-1$, if and only if $k_{n-j} = 0$ otherwise $k_{n-j} = 1$.

4.2.2 Goubin's Refined Power Analysis (RPA)

Goubin in [51] presents the RPA. The basic idea of this attack is to use "special points" P on the EC $E(\mathbb{K})$ in such a way that $P = (x, 0)$ or $P = (0, y)$. The attacker can choose the point input of $[k]P$ and to use for instance the base point $P = (c^{-1} \bmod \#E)(0, y)$ for some integer c, the point $[c]P = (0, y)$ leaving a significant difference in the consumption traces. The DPA can successfully detect the difference in the consumption traces. The attacker may know the secret key bits k recursively, thus this attack belongs to the family of attacks multiple execution. This means, that the attack can be applied to protocols that use the same private key k, for multiple executions, therefore the attack can be applied to single pass ECMQV where one of the ephemeral Diffie-Hellman/MQV keys is kept constant, ECIES and single pass ECD. Therefore for NIST and SECG curves over prime fields, there are only special points in the form $(0, y)$. An efficient countermeasure is the use of isogeny, discussed in Section 7 and countermeasures of several attacks are presented in Section 9.

4.2.3 Akashita and Takagi's Zero-value Point Attacks (ZPA)

Akashita and Takagi in [2] presents a generalization of Goubin's attack; the principal idea of ZPA is the use of special points, in such a way that the auxiliary register can take the zero-value, in particular for points $P = (x, y)$ which satisfies a) ED1: $3x^2 + a = 0$ or b) ED2: $5x^4 + 2ax^2 - 4bx + a^2 = 0$ that cannot be randomized by projective coordinates or random EC isomorphism or random field isomorphism. The attack depends of the addition formula implementation, particularly Akashita *et al.* uses the zero-value register for doubling point formula. Akishita in [2] recommended: *"In order to resist this type of attacks, we have be careful in implementing the addition formula"*. Similar to the above attack, an efficient countermeasure is the use of isogeny discussed in Section 7 and countermeasure of several attacks in Section 9.

Recently, Liu *et al.* in [86] performed an implementation of the *Montgomery Ladder* algorithm to protect against SSCA and Randomized Projective Coordinate to protect against DSCA, in particular they propose the algorithm Randomized MSB serial multiplication over $GF(2^m)$ that protects against ZPA.

4.2.4 Yen *et al.*'s C-safe Fault Attacks

This attack consists of introducing a fault in $[k]P$ at a point corresponding to a suspected dummy operation, if the final output is still valid, the guess was correct, whereas, if the fault produces an error in the final output, then the guess was incorrect. For instance, in the Algorithm 2, if an attacker disturbs the step 4 "R_1" and if a correct result is obtained at the end of the algorithm, then it is a "dummy" operation, hence $k_i = 0$, for the other case $k_i = 1$. For more details see [130].

4.2.5 Yen *et al.*'s M-safe Fault Attacks

The M-safe faults attack consist of apply a fault in some memory blocks which may be erased [131]. If we observe the Algorithm 2 performed in Step 4 $R_1 \leftarrow R_0 + P$, hence, if a fault is induced in R_0 once it has been used after the calculation of step 4, then if $k_i = 1$, the faults on line 5 in R_1 will be erased. Otherwise, if $k_i = 0$, then the value of R_0 is erroneous, this fault propagates to the end of the $[k]P$. Through this process the adversary can reveal k_i

4.2.6 The 2-Torsion Attack's (for fields of characteristic two)

Yen *et al.* in [129], introduces the 2-torsion attack. This is a SPA that uses a point of order 2 as input. In the context of EC an attacker uses the point P as input and observes the power consumption curves, hence there are only two possibilities: For example, *Double-and-add-always* Algorithm 2, the Step (5) of iteration i the register $R_0 = P_\infty$ if $k_i = 0$ or $R_0 = P$ if $k_i = 1$, then an attacker, by analyzing a single power consumption curve can know k_i bits of the secret key.

4.2.7 Correlation Collision attack on the horizontal setting

For further details see subsection 4.5.1.

Observation Table 11 shows a relationship between the types of attacks to *doubling-add-always* versus the requirements to perform the attack. We can see that most of the attack requires multiple executions.

4.3 Montgomery Ladder of Brier-Joye

The *Montgomery ladder algorithm* [94] (ML) was built for Montgomery EC defined over field of large characteristic. The ML algorithm for every bit of the k_i both operations an addition and a doubling are performed. With the supplementary condition that both operations have an impact on the final output of the $[k]P$. Later Brier and Joye [18] generalize this idea to Weierstrass curves defined over field of large characteristic. The computational cost for addition formula of ML EC is lower than the Weierstrass EC form, and its $[k]P$ is also faster. This was later generalized to all EC [18, 87, 52], and right-to-left scalar multiplication (*Double-add* of Joye's) [74]. The computational cost is $9M+2S$ for addition algorithm and $6M+3S$ for doubling algorithm. The classic ML is prone to M-safe attacks [131, 74]. Later Joye and Yen proposed a modification of the ML (Algorithm 3) in order to counteract M-safe fault attacks. Thus, the modified ML provides natural protection against SPA and safe-error attacks [74]. Notice that during the computing $[k]P$ using ML allows the use of x-coordinate only [18, 63, 44, 87]. The computational cost is: a) ML of Brier-Joye [18] is $n(12M+13S)+1I+3M+1S$. b) X-only ML [18, 63] is $n(9M+7S)+1I+14M+3S$. c) (X,Y)-only Co-Z ML [52] is $n(8M+6S)+1I+1M$. In Section 4.8 the lower computation cost of this countermeasure is presented.

Security Problems:

4.3.1 Twist Curve Fault Attacks x-only version

The x-only version of the ML is prone to twist curve fault attacks of Fouque [45] given that the twist curves $\widetilde{E}$ of many cryptographically strong curves could be smooth. Fouque *et al.* observed

Algorithm 3 Modified-Montgomery-ladder

Inputs: A point $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_0)_2 \in \mathbb{N}$
Outputs: $Q = [k] \cdot P$
1: $R_0 \leftarrow P_\infty$, $R_1 \leftarrow P$
2: **for** i **from** $n-1$ **to** 0 **do**
3: $\quad b \leftarrow k_i$,
4: $\quad R_{1-b} \leftarrow R_{1-b} + R_b$
5: $\quad R_b \leftarrow 2R_b$
6: **end for**
7: **return** R_0

that performing the $[k]P$ for some curve E of NIST [101] without using the y-coordinate, if the cryptosystem give us a correct result for $[k]P$ on $\widetilde{E}$ the attack is successfully applied only to just one or two faults during the computation, consequently any generic algorithm may resolve the discrete logarithm problem. Note that there are countermeasures for this attack, 1) Repeat point validity check during $[k]P$, 2) Use $y-$coordinate all the time, 3) Choose twist-secure curves.

4.3.2 RPA and ZPA

See Sections 4.2.2 and 4.2.3.

4.3.3 Relative Doubling Attack

The relative doubling attack is presented by Yen *et al.* in [128]. This attack uses the same chosen input as described in doubling attacks of Fouque *et al.* (P and $2P$)[40]. The attack deduces the existence of two equal adjacent bits in the secret scalar k i.e, it determines whether $k_i = k_{i-1} = 0$ or $k_i = k_{i-1} = 1$; the latter implies that the bit number k sought by the attacker is reduced.

4.3.4 Address-bit DPA (ADPA)

To more details see subsection 6.

4.3.5 Correlation Collision attack on the horizontal setting

To more details see subsection 4.5.1.

Observation Heyszl *et al.* in [59] presents an attack based on Electromagnetic (EM) analysis of temporal registers for ML algorithm over a field of characteristic two. The fundamental idea of the attack is to analyze the EM behavior in the temporal register location of the smart cards and thus, to know the bits of the secret key, since there is a direct relationship between the bits of the secret key and the temporary registers. Additionally, Heyszl presents a solution for the attack consisting of randomizing the location of the registers, thus avoiding a direct connection between the location of the registers and the bits k_i.

4.3.6 Attack using Unsupervised Learning

In [58] an attack over algorithms for $[k]P$ using ML and *double-and-add always* algorithms implemented in FPGA is presented. In particular this work focuses on attacking the ML algorithm

presented by Lopez-Dahab in [87] using projective randomized coordinates [24]. The attack uses unsupervised learning for single-execution side-channel leakage, exploit location-based position leakage [58] using EM [59]. The main idea is to divide a side-channel EM trace and to use the k-means clustering algorithm. The attack depends on the ability to acquire multiple EM traces simultaneously of different probe positions. In this way, it identifies for each subtrace t_i the detected location-based leakage depends on the measures position on the surface on the die [58] that depends directly on the bits k_i.

On the other hand, Perin in [106] uses an unsupervised attack since it does not require previous knowledge of the device to be attacked, for the attack it uses four phases: trace pre-processing, points of interest identification, fuzzy k-means clustering, and exponent recovery. For the final stage they use statistical tools: majority rule, density probability function and Bayesian classifier, where the side-channel data is captured from an EM measures in an FPGA. The attack is applied to RSA using ML, *exponent blinding* as a countermeasure for DSCA ($d_r = d + r * \phi(N)$) and contrameasures to protected with the *Leak Resistant Arithmetic* [8]. The authors mention that the attack can be apply to ECC, capturing all bits k_i.

Later Specht *et al.* in [119] improves the results obtained by Heyszl *et al.* [58] apply Principal Component Analysis (PCA), expectation maximization clustering-based and simple pre-processing used by Perin. This attack is non-profiled for single-execution trace against on $[k]P$ implemented on an FPGA.

4.3.7 Cmov Side Channels

Nascimento *et. al.* in [100] presents a new attack over ML algorithm. The attack initially uses a *Template Attacks* to study two different attacks techniques, the first is based on the study of *conditional swaps* (cswaps), here we can observe that the behavior of these temporal registers depend directly on the bits of the secret key k_i, its attack has an 96.71% effectiveness. The second idea can also be carried with *secret-dependent memory accesses*, for more details see [100].

4.4 Double-Add of Joye

The *Joye's double-add algorithm* in [73] is a ML Algorithm for *Right-to-left*. The Algorithm 4 shows the Joye's *double-add* resistant against SPA. The computational cost is: a) Classic Joye's double-add [73] is $n(13M+8S)+1I+3M+1S$. b) Co-Z Joye's double-add [52] is $n(9M+7S)+1I-9M-6S$. In Section 4.8 the lower computation cost of this countermeasure is presented. For this countermeasure there are not attacks reported in the literature.

Algorithm 4 Joye's double-add resistant against SPA

Inputs: A point $P \in E(\mathbb{F}_q)$ and $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Outputs: $Q = [k] \cdot P$
1: $R_0 \leftarrow P_\infty$, $R_1 \leftarrow P$
2: **for** i **from** 0 **to** $n-1$ **do**
3: $\quad b \leftarrow k_i$,
4: $\quad R_{1-b} \leftarrow 2R_{1-b} + R_b$
5: **end for**
6: **return** R_0

4.5 Joye's Add-only algorithm

The *Joye's add-only algorithm* was presented in [73]. The Algorithm 5 shows the Joye's add-only resistant against SPA. The computational cost is $(2n)A$.

Algorithm 5 Joye's Add-Only Scalar Multiplication

Inputs: $P \in E(\mathbb{F}_p)$ and n-bit scalar $k = (k_{n-1}, k_{n-2}, \ldots, k_0)_2$
Outputs: $Q = [k]P$
1: $R_0 \leftarrow P_\infty$, $R_1 \leftarrow P$, $R_2 \leftarrow P$
2: **for** i **from** 0 **to** $n-1$ **do**
3: $\quad R_{1-k_i} \leftarrow R_{1-k_i} + R_2$
4: $\quad R_2 \leftarrow R_0 + R_1$
5: **end for**
6: **return** R_0

Security Problems:

4.5.1 Correlation collision attacks in the horizontal setting

Hanley *et al.* in [57] presents an improved technique to detect internal collisions used to apply SCA. The attack is applied in two platforms, using ARM7TDMI software and SASEBO-G FPGA hardware [110]. Hanley, notes that the register R_0 remains the same if the bit $k_{i+1} = 0$ and the register R_1 is the same if $k_{i+1} = 1$.

The attacker uses these collisions to know the bits of the secret key k_i. In particular, he apply the attack by observing collisions between the second addition in the loops which operates the bit k_i and the first addition in the next loop that operates with bit k_{i+1}, this occurs if bit $k_{i+1} = 1$. In order to find collisions, Hanley *et al.* uses two approaches: the Pearson correlation coefficients and Euclidean distance. Additionally, Hanley *et al.* presents a modified attack to countermeasures Coron's *Double-add-Always* and ML.

In the case of *Double-add-Always*, since the algorithm for each bit k_i of the secret key executes a doubling $R_0 \leftarrow 2R_0$ and an addition given by $R_{1-k_i} \leftarrow R_0 + P$, the comparison of these operations is based on the study of collisions at the multiplication level on the field ($2R_0$ and $R_0 + P$). For further details see [57].

For ML algorithm, for each bit executes $R_{\neg k_i} \leftarrow R_{k_i} + R_{\neg k_i}$ and $R_{k_i} \leftarrow 2R_{k_i}$ Hanley notes the following collisions: (a) *"If the bits treated in two consecutive loops are the same then the output of the operation in* $R_{k_i} \leftarrow 2R_{k_i}$ *in the first loop will be the input to the operation in* $R_{k_i} \leftarrow 2R_{k_i}$ *on the second loop."* (b) *"If the bits treated in two consecutive loops are different then the output of the operation in* $R_{\neg k_i} \leftarrow R_{k_i} + R_{\neg k_i}$ *in the first loop will be the input to the operation in* $R_{k_i} \leftarrow 2R_{k_i}$ *on the second loop."*

This attack is not used to directly compare operations, since the addition and doubling consists of different operations. Moreover, one cannot compare field operations directly since one wishes to compare the input of one operation with the output of another operation.

Recently, a countermeasure to protect the latter attack is proposed in [85], see Algorithm 6.

4.6 Signed Digit Methods Goundar

In order to prevent SPA-type attacks, the *zeroless signed-digit expansion* (ZSD) is considered. Lets be an odd integer k then we can express this with digits {-1,1}, the idea was presented by

Algorithm 6 Randomized Montgomery Ladder

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$, and $k_{n-1} = 1$
Outputs: $\mathbf{Q} = [k] \cdot \mathbf{P}$

$b \xleftarrow{Random} \{0,1\}$, $R_0 \leftarrow P_\infty$
if $b = 0$ **then**
  $R_1 \leftarrow P$
else
  $R_1 \leftarrow P_\infty$
end if
for i **from** $n-1$ **to** 0 **do**
  if $b \oplus k_i = 1$ **then**
    $b \xleftarrow{Random} \{0,1\}$
  end if
  $R_b \leftarrow R_0 + R_{b \oplus k_i}$
  $R_{\neg b} \leftarrow R_b + P$
end for
return R_0

Goundar *et al.* in [52]. The Algorithm 7 show the *Signed-digit* method for *Left-to-right*. The computational cost is: a) Co$-Z$ signed-digit algorithm (Right-to-left) [52] is $n(9M + 7S) + 1I - 9M - 6S$. b) $(X, Y)-$only co$-Z$ signed-digit algorithm (Left-to-right) [52] is $n(8M + 6S) + 1I - 5M - 4S$. In Section 4.8 the lower computation cost of this countermeasure is presented. For this countermeasure there are no attacks reported in the literature.

Algorithm 7 Signed-digit method: Left-to-right

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$ with $k_0 = 1$
Outputs: $\mathbf{Q} = [k] \cdot \mathbf{P}$

$R_0 \leftarrow P$; $R_1 \leftarrow P$
for i **from** $n-1$ **to** 1 **do**
  $\kappa \leftarrow (-1)^{1+k_i}$
  $R_0 \leftarrow 2R_0 + (\kappa)R_1$
end for
return R_0

4.7 Atomic Blocks Chevallier-Manes

The Atomic Blocks idea was presented by Chevallier-Manes *et al.* [20] and consists of pulling away the field operations of Addition and Doubling in small homogeneous atomic blocks, where they are not distinguishable from each other through SSCA. This atomic block had a structure of *Multiplication-Addition-Negation-Addition* operations (M, A, N, A) over the prime field and made an assumption that $M = S$ from a side-channel perspective. Later Hanley *et al.* [54] and Amiel *et al.* [4] refuted the latter assumption.

This distinction may have some efficiency benefits, since squaring is less expensive than multiplication [47]. A flexible methodology was introduced by Bernstein *et. al.* [12] and Longa *et. al.* [82] where it is possible to prove that is very useful. This methodology enables modifying the addition and doubling operations to balance the number of S and M, thus facilitating the introduction of squaring into atomic blocks, here the atomic block structure is (S, N, A, M, N, A, A).

Abarzúa and Thériault, to improve the security aspects of previously published atomic blocks in [1], these atomic block are designed as countermeasures against both SSCA and C-

safe fault attacks for scalar multiplication. The atomic block structure follows the sequence (S, N, A, A, M, A).

4.7.1 Point Doubling in Jacobian Coordinates.

Let $P = (X_1 : Y_1 : Z_1)$ be a point in Jacobian coordinates on the EC E. The Doubling Algorithm requires $4M + 4S$ (to more detail see [1]). Table 1 shows the atomic blocks for doubling, taking as input $R_1 \leftarrow X_1$, $R_2 \leftarrow Y_1$, and $R_3 \leftarrow Z_1$, and returning as output $X_3 \leftarrow R_1$, $Y_3 \leftarrow R_2$, and $Z_3 \leftarrow R_3$.

Table 1: Atomic block formula for Jacobian doubling

Sec	Block 1	Block 2	Block 3	Block 4
S	$R_4 \leftarrow R_3^2$ $[Z_1^2]$	$R_6 \leftarrow R_2^2$ $[Y_1^2]$	$R_4 \leftarrow R_1^2$ $[\alpha^2]$	$R_8 \leftarrow R_7^2$ $[4Y_1^4]$
N	$R_5 \leftarrow -R_4$ $[-Z_1^2]$	$R_7 \leftarrow -R_1$ $[-X_1]$	$R_5 \leftarrow -R_1$ $[-\alpha]$	$R_2 \leftarrow -R_8$ $[-4Y_1^4]$
A	$R_6 \leftarrow R_1 + R_4$ $[X_1 + Z_1^2]$	$R_1 \leftarrow R_7 + R_7$ $[-2X_1]$	$R_8 \leftarrow R_6 + R_6$ $[-2\beta]$	$R_8 \leftarrow R_1 + R_6$ $[X_3 - \beta]$
A	$R_4 \leftarrow R_1 + R_5$ $[X_1 - Z_1^2]$	$R_7 \leftarrow R_6 + R_6$ $[2Y_1^2]$	$R_1 \leftarrow R_4 + R_8$ $[X_3 = \alpha^2 - 2\beta]$	$R_4 \leftarrow R_2 + R_2$ $[-8Y_1^4]$
M	$R_5 \leftarrow R_6 R_4$ $[X_1^2 - Z_1^4]$	$R_6 \leftarrow R_1 R_7$ $[-\beta]$	$R_4 \leftarrow R_2 R_3$ $[Y_1 Z_1]$	$R_6 \leftarrow R_5 R_8$ $[-\alpha(X_3 - \beta)]$
A	$R_4 \leftarrow R_5 + R_5$ $[2(X_1^2 - Z_1^4)]$	$R_1 \leftarrow R_5 + R_4$ $[\alpha]$	$R_3 \leftarrow R_4 + R_4$ $[Z_3 = 2Y_1 Z_1]$	$R_2 \leftarrow R_6 + R_4$ $[Y_3]$

4.7.2 Mixed Addition in Jacobian-Affine Coord.

Given the points $P = (X_1 : Y_1 : Z_1)$, in Jacobian coordinates, and $Q = (X_2, Y_2)$, in affine coordinates, both on the EC E. The mixed addition formula $P + Q = (X_3 : Y_3 : Z_3)$ requires $8M + 3S$. The resulting atomic blocks can be seen in [1].

The computational cost for this countermeasure [1] is $n(7M + 7S) + 1I + 3M + 1S$ [2]. In Section 4.8 the lower computation cost of this countermeasure is presented.

Security Problems:

The atomic blocks presented by [1] are prone to Horizontal Collision Correlation attack proposed by [9] see Subsection 4.1.5. Moreover, Murdica [97] in his doctoral thesis presents a Vertical Collision Correlation attack based on the Bauer Attack, see Subsection 4.1.6. Additionaly, this atomic block are prone to Fouque's Doubling attacks [40] see Subsection 4.2.1 and Chen's attack [26][3].

4.8 Summary Performance & Security Problems Countermeasures of SSCA in ECC

The binary representation of $k = (k_{n-1}, \ldots, k_0)_2$. Tables 2 and 3 show a summary of the lower computational cost, an algorithm description and the security problems of the different countermeasures for SSCA in ECC.

[2] We consider this ratios for our computing cost and $S/M = 0.8$.

[3] This experimental attack is applied because the implementation does not prevent irregular breaks between atomic blocks within the same group operation and distinct group operations.

Description:

The next items represent the algorithms which have been evaluated. Observe that we focus on e most efficient algorithms that exist in the literature for each one of the different countermeasures families. (a) : Using Fast Mixed Addition $(7M + 4S)$ and Fast Point Doubling $(3M + 5S)$ with $(a = -3)$, in [82]. (b) : (X, Y)-only co-Z Montgomery ladder, $(8M + 6S$ for each bit) in [52]. (c) : $X-$only Montgomery ladder, $(9M + 7S$ for each bit) in[18, 63]. (d) : $(X, Y)-$only co-Z signed-digit algorithm $(8M + 6S$ for each bit), in [52]. (e) : For addition $6M + 6S$ and doubling $4M + 4S$, in this case the algorithm performs $nD + \frac{n}{2}A$, in [1].

Table 2: *Left-to-right:* Comparison of the different regular multiplication algorithm

Countermeasure	**Coord. Sys.**	**Total Cost**	**Performance** $n = 192$	**Security Problem**
Unified Formulae	$\mathcal{P}$	$n(13M + 5S) + 1I + 2M$	$3366M$	(ψ)
Weierstrass curves		$n(16M + 3S) + 1I + 2M$	$3634.8M$	(ϕ)
Double-and-Add-Always	$\mathcal{J}$	$n(10M + 9S) + 1I + 3M + 1S^{(\mathbf{a})}$	$3406.2M$	(φ)
Montgomery Ladder	$\mathcal{J}$	$n(8M + 6S) + 1I + 1M^{(\mathbf{b})}$	$2558.6M$	(τ)
Weierstrass curves		$n(9M + 7S) + 1I + 14M + 3S^{(\mathbf{c})}$	$2919.6M$	(ϱ)
Signed-digit algorithm	$\mathcal{J}$	$n(8M + 6S) + 1I - 5M - 4S^{(\mathbf{d})}$	$2549.4M$	
Atomic Blocks	$\mathcal{J}$	$n(7M + 7S) + 1I + 3M + 1S^{(\mathbf{e})}$	$2523M$	(ξ)

Attacks summary:

The following items represents the different attacks for SSCA contrameasures in ECC. (ψ) : Izu and Takagi attacks [63], Walter attacks [127], Amiel attacks [4], Combined attacks [118], Bauer attack [9], Horizontal SVA [34]. (ϕ) : Stebila and Thériault attacks [115], Amiel attacks [4], PACA [5]. (φ) : Safe-error analysis C-type [130] and M-type [131], Fouque's Doubling attacks [40], RPA [51], ZPA [2], 2-torsion attacks (only fields characteristic two) [129], Correlation Collision Attack on horizontal setting [57]. (τ) : Relative Doubling attacks [128], Address-bit DPA [65], RPA [51] y ZPA [2], Correlation Collision Attak on horizonal setting [57] [59], Attack using Unsupervised learning [58], Cmov Side Channels [100]. (ϱ) : Twist curve fault attacks the Fouque [40]. (ξ) : Fouque's Doubling attacks [40], Chen's attacks [26], Horizontal Collision Correlation attack [9], Murdica [97].

Table 3: *Right-to-Left:* Comparison of the different regular multiplication algorithms

Countermeasure	**Coordinate Systems**	**Total Cost**	**Performance** $n = 192$	**Security Problem**
Joye's double-add	$\mathcal{J}$	$n(9M + 7S) + 1I - 9M - 6S^{(\mathbf{f})}$	$2889.4M$	
Signed-digit algorithm	$\mathcal{J}$	$n(9M + 7S) + 1I - 9M - 6S^{(\mathbf{g})}$	$2889.4M$	
Atomic Blocks	$\mathcal{J}$	$n(8, 5M + 8, 5S) + 1I + 3M + 1S^{(\mathbf{h})}$	$3041.4M$	(χ)
Add-only Joye	$\mathcal{J}$	$2n(7M + 4S) + 1I + 3M + 1S^{(\mathbf{i})}$	$4020.6M$	(ω)

Description:

(f) : Co-Z Joye's double-add, $(9M + 7S)$ for each bit, in [52]. (g) : Co-Z signed-digit algorithm, $(9M + 7S)$ for each bit, in [52]. (h) : For general addition $9M + 9S$ and doubling $4M + 4S$, in this case the algorithm performs $nD + \frac{n}{2}A$, in [1] (i) : Using Fast Mixed Addition $(7M + 4S)$.

Attacks summary:

(χ) : Chen's attacks [26]. (ω) : Correlation collision attack [57]. Note that the most efficient countermeasures for SSCA in ECC are (X, Y)-only Co-Z Montgomery ladder, Co-Z Signed-digit algorithm and Atomic Blocks.

5 Countermeasure for Differential Power Analysis in ECC

Differential Side-channel Analysis (DSCA) [78] uses statistical tools to recover the k_i bits on the secret key, based on the measurements from several $[k]P$. Brier *et al.* in [10] presents an improved DSCA since it requires fewer curves for recovering the key in contrast with the original DSCA. Recent results presented by [105, 43, 120] improve the attack.

Sets of Countermeasures Randomization of the Scalar This family is considered an effective countermeasures against RPA [51], and ZPA [2] if it is used with random base point [53].

5.1 Coron's First Countermeasure [24]

Let $\#E$ be the order of E. The computing $Q = [k]P$ is done by the following steps: a) Select a random number d of size n bits. Coron consider $n = 20$ bits. b) Compute $k' = k + d(\#E)$. c) Compute the scalar multiplication $Q = [k']P = [k + d(\#E)]P = [k]P + [d(\#E)]P = [k]P$, since $[d(\#E)]P = P_\infty$. In Table 4 the average loss cost is presented.

Table 4: Theoretical loss Cost

	NIST curves				
n bits of d	P−192	P−224	P−256	P−384	P−521
20-bits	10.4%	8.9%	7.8%	5.2%	3.8%
32-bits	16.6%	14.2%	12.5%	8.3%	6.1%
40-bits	20.8%	17.8%	15.6%	10.4%	7.6%

Security Problems:

5.1.1 Okeya and Sakurai Analysis

Okeya *et al.* in [102] analyzed the first countermeasure of Coron's. The authors analyze the existence of some relation of k' which depends only on the secret key k (in the least significant bits). Okeya studied the different possibilities of the integer d and computing the probabilities of different values k', showing that an attacker is able to derive information on the secret key k by statistical analysis of output k'. Okeya notes that this attack is effective even when the device is immune to SPA.

5.1.2 Fouque's Doubling Attacks

Fouque *et al.* in [40] presented a weakness of this countermeasure, due to the birthday paradox, after 2^{10} runs of P and $2P$ respectively, it is possible obtain a scalar k in such a way that there is a collision. The attacker needs to compare each curve obtained with P and $2P$. The criteria used is as follows: *"If two measurements have many common intermediate squaring, then a correct pair is found"*, hence, to identify the right pair a set of 2^{20} comparisons is required. For more details see [40].

5.1.3 Ciet and Joye Analysis

Ciet and Joye in [21] presented the following analysis for this countermeasure. In NIST and SECG, ECC defined over a prime field $\mathbb{F}_p$, p is the form $p = 2^l \pm 2^m \pm 1$ where m is a relatively small number. As a result, from the Hasse theorem, we have $|\#E - p - 1| \leq 2\sqrt{p}$ therefore it follows that the binary representation of $\#E$ is likely which is constituted by 1 and a long run of 0's. For example, in hexadecimal notation the EC "secp224k1" is: $\#E = (01000000000000000000000000001DCE8D2EC6184CAFOA971769FB1F7)_{16}$ points. The randomized multiplier k' typically looks as: $k' = k + d(\#E) = (d)_2||k_{l-1} \cdots k_{l-t}||$some bits. Notice that a set of bits of the secret scalar can be found by an attacker after the randomization. Then, applying DSCA $[k']P = [k + d(\#E)]P$ it is possible to recover the t most significant bits of scalar k.

5.1.4 Fouque's Carry Leakage

Fouque *et at.* in [46] presented the next idea, this attack analyze the behavior of the device when computing the secret scalar of the sum of $k + d(\#E)$. Nowadays, the devices has $l-$bit architectures where l is a multiple of 8, then Fouque *et at.* demonstrated that the behavior of the partial sums of the carry out $c_i = k_i + d_i(\#E)$ (with $i = 0, \ldots, k-1$) dependent directly of secret key k_i and not on the bits of mask $d_i(\#E)$ (hence it is assumed that the attacker can deduce the c_i of carries output).

5.1.5 Feix attack

Feix *et al.* in [43] observes that the order of NIST standard curves are separated into three categories. Type-1: the order has a large pattern of ones. Type-2: the order has a large patterns of zeros, Type-3: the order has a combination large of zeros and ones. Given the form of the orders, the mask $d(\#E)$ also has a specific representation. The attack consists of three parts: (a)*First a vertical Collision-correlation attack is performed in order to find the bits of scalar k which are not masked through $d(\#E)$.* (b)*The second part is to find the random part of k' that appears in the most significant bits part of k' and a horizontal correlation is applied.* (c)*As in the two previous parts of the attack, the most significant bits of the private key k and the blinding of k' are known, in the third part the least significant bits of the secret part is recovered given that the random value d is known. Here a classic vertical correlation attack is used.*

5.1.6 Big Mac Attack

This countermeasure does not protect against the Big Mac attack in Section 8.

5.2 Clavier-Joye Countermeasure

Clavier *et al.* in [22] presented the *Exponent Splitting*, where for any random number r of $n-$bit, i.e. the same length of the secret key k, it is computed by: $[k]P = [k-r]P + [r]P$. To generate a random number r is expensive; this countermeasure requires at least two procedures both $[k-r]P$ and $[r]P$.

Security Problems:

5.2.1 Ebeid Analysis

Ebeid in [36] studied the implementations of this countermeasure using the *Shamir-Strauss's trick* algorithm [121] and found internal collisions that constitute a vulnerability which can be attacked by a DPA; for this countermeasure, Ebeid studies each term of $[k-r]P$ and $[r]P$ and recommend that it will be should computed separately using a SPA-resistant algorithm.

5.2.2 Muller and Valette Attack's

The basic idea of this attack, presented by [91], is to study the statistical properties of exponent splitting at the bit level of $[k-r]$ and $[r]$. The pair $([k-r],[r])$ is not uniformly distributed, since it always satisfies $[k-r]+[r] = [k]$. Muller *et al.* shows that there is a bias in the distribution of the i-th bits of the pair $([k-r],[r])$. At the bit level, the following relation is satisfied $c_i \oplus r_i \oplus (k-r)_i = k_i$, where, r_i, $(k-r)_i$, k_i and c_i, respectively denotes the i-th bits of r, $(k-r)$, k and carry bit c in the addition $[r]+[k-r]$. The analysis of Muller and Valette studies the transition probabilities obtained from c_i and c_{i+1} for different $k_i = \{0,1\}$, and $([r_i],[k-r]_i), = \{(0,0),(0,1),(1,0),(1,1)\}$. Muller's study can be seen as the two bits $([r_i],[k-r]_i)$ are not uniformly distributed and the imbalance depends on the bits value. Actually, Muller *et al.* use Markov chain to bit-level and the probability transition rules in order to derive the step i from the step $i-1$, for more detail see [91].

5.2.3 Fouque *et al.* Carry Leakage Attack

Applying the ideas studied by section 5.1.4, it is observed that this attack can be performed to this countermeasure, since the analysis can be carried out for $-r$ instead of $d\#E$, which is applied by Fouque in [46].

5.2.4 Ha *et al.* Analysis Using the 2-Torsion Attack's (only for fields of characteristic two)

Ha *et al.* in [53] introduced the next analysis using 2-Torsion Attacks of Yen [129]. Suppose an attacker can find the 2-torsion point P and uses this point as input to compute $[r]P$, therefore $[r]P$ can be computed with other power attack countermeasures, such as the BRIP (Algorithm 15) or *doubling-and-add-always* (Algorithm 2), the attacker can derive a secret random number r and $k-r$ in two independent $[k]P$ using the 2-torsion attack. Furthermore, it can be easily avoided by checking $2P \neq P_\infty$ before computing $[k]P$ in [92].

5.2.5 Big Mac Attack

The countermeasure does not protect against Big Mac attack presented in section 8.

5.3 Trichina-Bellezza, Countermeasure

Trichina *et al.* in [125] proposed the following countermeasure, for any random number r to compute: $[k]P = [kr^{-1}]([r]P)$. The principal disadvantage of this countermeasure is to compute the inverse of r module $\text{ord}_E(P)$. Furthermore, two scalar multiplication are needed, first $R = [r]P$ and later $[kr^{-1}]R$ is computed. Trichina *et al.* in [125] indicates that "*one way for this countermeasure to be efficient is to keep the overhead low, one can choose r to yield a fast* $[r]P$*, for instance by choosing r among the elements of* $\mathbb{F}_p$ *of at most t-bits, with a small t*". This contrameasure does not present security problems.

5.4 Ciet-Joye, Countermeasure

In [21] a *random key splitting* is presented and is called the *Euclidean division*, that is, k is written as: $[k]P = [k \bmod r]P + [\lfloor k/r \rfloor]([r]P)$. Letting $S := [r]P$, $k_1 := k \bmod r$ and $k_2 := \lfloor k/r \rfloor$ we can obtain $Q = [k]P = [k_1]P + [k_2]S$ where the bit length of r is $n/2$. The next algorithm describes a regular variant of *Shamir's double ladder*. We let l denote the bit-length of $\max(k, d)$ –and thus k_{l-1} and d_{l-1} are equal to 1. As we can be seen, the Algortihm 8 requires calculate only addition and doubling for each bit, that is, it has the same complexity as the algorithm *"double and add always"*.

Algorithm 8 Regular variant of Shamir's double ladder $Q = [k]P + [d]S$

Inputs: Point $\mathbf{P}$ and $\mathbf{S} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_0)_2$, and $d = (d_{n-1}, \ldots, d_0)_2 \in \mathbb{N}$,
Outputs: $\mathbf{Q} = [k]\mathbf{P} + [d]\mathbf{S}$
1: $R_1 \leftarrow P$; $R_2 \leftarrow S$; $R_3 \leftarrow P + S$; $c \leftarrow 2d_{l-1} + k_{l-1}$; $R_0 \leftarrow R_c$
2: **for** i **from** $n-2$ **to** 0 **do**
3: $\quad R_0 \leftarrow 2R_0$
4: $\quad b \leftarrow \neg(k_i \vee d_i)$; $c \leftarrow 2d_i + k_i$; $R_b \leftarrow R_b + R_c$
5: **end for**
6: **return** R_0

Security Problems:

5.4.1 Ebeid Analysis

Ebeid in [36] studied the above countermeasure and identified the occurrence of some collisions on intermediate points. Ebeid recommends a way to avoid these collisions, which is to make the quotient $\lfloor k/r \rfloor$ always odd. That means that if $\lfloor k/r \rfloor$ is even, it is decreased by one and $[k \bmod r]$ is updated by adding r to it. This may increase the bit length of $[k \bmod r]$ to $l + 1$.

5.4.2 Ha *et al.*'s Analysis using 2-Torsion Attack

Ha *et al.* in [53] introduced the next analysis. Suppose an attacker can find the 2-torsion point G and using this point to input, the attacker can derive a secret random number r that is detected during the computation of $S = [r]P$ using 2-torsion attack. Also, k/r and $(k \bmod r)$ are detected during the computation of $\lfloor k/r \rfloor S + (k \bmod r)P = \lfloor k/r \rfloor P + (k \bmod r)P$, because all intermediate values are 3 types P, $2P$ or P_∞ when r is odd. If r is even and input point is $G = 2P = P_\infty$ then $S = [r]P = [r]G = P_\infty$. The $(k \bmod r)$ are also detected during the computation of $(k \bmod r)P$.

5.5 Chevallier-Mames's Self-Randomized Exponentiation Algorithms

Chevallier-Mames in [23] presented the use of the *exponent splitting method*. The idea is as follows: let $k = (k_l, \ldots, k_0)_2 = \sum_{i=0}^{l} k_i 2^i$ with $k_i \in \{0, 1\}$ denote the binary representation of scalar k and defining: $k_{d \to j} := (k_d, \ldots, k_j)_2 = \sum_{j \leq i \leq d} k_i 2^{i-j}$. *Left-to-right* Algorithm 1, share the common feature that an accumulator is used throughout the computation for storing the value of $[k_{l \to i}]P$ for decreasing i's until the accumulator contains the value of $Q = [k_{l \to 0}]P$. The principal idea of this countermeasure (Algorithm 9) is taking part of k as a source of randomness. The algorithm relies on the simple observation that, for any $0 \leq i_j \leq l$, we have $[k]P = [k_{l \to 0}]P = [(((k_{l \to 0} - k_{l \to i_1}) - k_{l \to i_2}) - k_{l \to i_3}) \cdots - k_{i_f}]P + [k_{l \to i_1}]P + [k_{l \to i_2}]P + [k_{l \to i_3}]P + \cdots + [k_{l \to i_f}]P$. If the i_j's are randomly chosen, the $[k]P$ algorithm becomes probabilistic. A Boolean random variable ρ is used to determine whether or not the current loop index i belongs to the set $\{i_1, \ldots, i_f\}$.

To ensure the correctness of the process, the randomization step $k \leftarrow k - k_{l-i_j}$ cannot modify the $(l - i_j + 1)$ most significant bits (i.e. $k_{l \to i_j}$) of k. This condition is guaranteed by checking that $k_{l \to i_j} \leq k_{i_j - 1 \to 0}$. Furthermore, the consistency condition i.e., $k_{i_j - 1 \to 0} \geq k_{l-1 \to i_j}$ implies that only the lower half of exponent k is randomized. The performance loss is $10A$ for a curve to P−192, for details see Alg. II in [23]. This countermeasure does not present security problem.

Algorithm 9 Self-randomized exponentiation: Left-to-right

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{l-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Outputs: $\mathbf{Q} = [k] \cdot \mathbf{P}$

```
R0 ← P∞; R1 ← P∞; R2 ← P; i ← l − 1
while (i ≥ 0) do
    R0 ← R0 + R0
    if (ki = 1) then
        R0 ← R0 + R2
    end if
    ρ ← Random{0, 1}
    if ((ρ = 1) ∧ (ki−1→0 ≥ kl−1→i)) then
        k ← k − kl−1→i
        R1 ← R1 + R0
    end if
    i ← i − 1
end while
R0 ← R0 + R1
return R0
```

5.6 Summary Performance & Security Problems Countermeasures DPA Randomization of the Scalar

The next items represent the algorithm which have been evaluated. Observe that we focus on the most efficient algorithm that exist in the literature for each one of the different countermeasures families. (a) High: $\approx 100\%$, Medium: $(30 - 70)\%$, Low: $(10 - 25)\%$, Negligible: $< 0.5\%$. (b) On average performance loss is 15.9% for the curve P−192. (c) To avoid opening the way to new attacks, $[k - r]P$ and $[r]P$ must be computed separately, doubling the cost of the scalar multiplication (Ebied in [36]). (d) Two scalar multiplication and one inversion are needed. (e) Two scalar multiplication are needed. (f) Performance loss is $10A$ for a curve to P−192, for details see Alg. II in [23].

Attacks summary:

Table 5: Comparison of the Different DPA Countermeasures Randomization of the Scalar

Countermeasure	**Computation Overhead**$^{(a)}$	**Security Problem**
Coron's First Countermeasure [24]	Low$^{(b)}$	ϕ
Clavier-Joye Counter. Exp. Splitting [22]	High$^{(c)}$	φ
Trichina-Bellezza Countermeasure [125]	High$^{(d)}$	
Ciet-Joye Countermeasure [21]	Medium$^{(e)}$	χ
Chevallier-Mames Self-Rand. Expo. [23]	Negligible$^{(f)}$	

ϕ: Okeya and Sakurai Analysis [102], Fouque's Doubling Attacks [40], Ciet and Joye Analysis [21] (just for secp224k1 curve), Fouque's Carry Leakage [46], Feix attack [43], Big Mac Attack [127]. φ: Ebeid Analysis [36], Muller and Valette Attack's [91], Ha Analysis Using the 2-Torsion Attack's (just for fields of characteristic two) [53], Fouque Carry Leakage [46], Big Mac Attack [127]. χ: Ebeid Analysis [36], Ha *et al.*'s Analysis using 2-Torsion Attack in [53]

Set of Countermeasures Randomization Point In this section we will study the countermeasures knowns as Randomization Point. The countermeasures and their security problems will be presented below.

5.7 Blinding the Point Second Countermeasure of Coron's

In [24] the next idea is presented; for scalar multiplication $[k]P$, firts $[k](P+R)$ is compute and at the computation end $S=[k]R$ is subtracted. This countermeasure is effective against RPA, ZPA and SVP, given that the attacker cannot freely choose the base point. See section 7. This countermeasure is considered inefficient, since it must perform two scalar multiplications $S=[k]P$ and $[k](P+R)$.

Algorithm 10 Coron's Blinding Point Second Countermeasure

Inputs: Point $\mathbf{P}$ and secret point $\mathbf{R} \in E(\mathbb{F}_q)$,
$k=(k_{l-1},\ldots,k_1,k_0)_2 \in \mathbb{N}$, and $S^{'}=[k]R$
Outputs: $\mathbf{Q}=[k]\cdot\mathbf{P}$
1: $P \leftarrow P+R$;
2: pick$\beta \in \{0,1\}$at random
3: $R \leftarrow (-1)^{\beta}2R$
4: $S^{'} \leftarrow (-1)^{\beta}2S^{'}$
5: $S=$ double-and-add$(P,[k])$
6: **return** $S-S^{'}$

Security Problems:

5.7.1 Okeya and Sakurai Analysis

Okeya *et al.* in [102] presented the next analysis. Let $P, 2P, 4P, \ldots, 2^kP$ be a point on the EC, and $C_j(t)$ be a function of power consumption associated with the execution of $[k](2^jP)$. First, an attacker feeds these points to the cryptographic device that is equipped with this countermeasure. Then, the attacker gets the functions $C_j(t)$ and calculates the correlation function by: $g(t)=\frac{1}{n}\sum_{j=0}^{n-1}\min\left[\frac{1}{(C_j(t+t_0)-C_{j+1}(t))^2},M\right]$. Where t_0 is the time required for each

round, which must be constant to counteract a SPA. M is some large constant, if the function $g(t)$ tends to infinity.

Okeya assumes that an attacker can take some points from one execution to another execution. This attack analyze the behavior of the correlation function $g(t)$ and notes it has a strong relationship with the bits of the secret key k. He concludes: *If* $k_i = 1$ *then* $g(t)$ *by vanishing and* $k_i = 0$ *then* $g(t)$ *is not vanishing.* Through this analysis an attacker can find the bits of the secret key k. Besides, a countermeasure for this attack is presented called *Refresh procedure* of the points R and S (see [102] for details).

5.7.2 Fouque's Doubling Attacks

Fouque *et al.* in [40] presented an analysis for this countermeasure. As input to the micro-controller a point P is required, then the micro-controller executes $P + R$. The adversary then requests the computation with the point $2P$. With probability $1/2$, the micro-controller will use the point $2P + 2R = 2(P + R)$. So, the attacker compares two side channel measurements and to recover the secret scalar k. Fouque shows that if the noise is too strong, the adversary can use a statistical approach in order to find the secret scalar. The attacker uses a random point Q and compute $[k]Q$ and $[k]2Q$ in order to analyze the difference between the first and the second curve using doubling attack.

5.7.3 Big Mac Attack

This countermeasure does not protect against the Big Mac attack in section 8.

5.8 Third Countermeasure of Coron's, Randomized Projective Coordinates

The third countermeasure of Coron [24] called *Randomizing the Homogeneous Projective coordinates* of point $P = (X, Y, Z)$ with a random $\lambda \neq 0$ to $P = (\lambda X, \lambda Y, \lambda)$. The random value λ can be updated in every execution or after each addition-doubling. When, $[k]P$ is computed using Jacobian coordinates, the point $Q = [k]P$ is represented as $Q = (X, Y, Z)$. So, to avoid the attack presented by [99] the point Q must be recovered to affine coordinate by computing $x = X/Z^2$ and $y = Y/Z^3$, this attack is presented in Section 5.8.2. Moreover, a Jacobian coordinate and the curve parameter $a = -3$ is suggested. Using this technique is much more efficient, but does not allow $\lambda = 1$ (for detail see [122]), i.e. in scalar multiplication, we cannot use mixed coordinates (Jacobian and affine). Additionally, this countermeasure is effective against Template Attacks [19]. This countermeasure has a very low computing cost, here $3M$ for the homogeneous representation and $4M + 1S$ for the Jacobian representation are required.

Security Problems:

5.8.1 RPA [51], ZPA [2], SVA [96]

See subsections 7.1, 7.2 and 7.3.

5.8.2 Projective Coordinates Leak

Naccache *et al.*'s in [99] observed the following analysis of this countermeasure. Let the point Q be an point of prime order in E over prime fields $\mathbb{F}_p$. Denote the $Q = (x_Q, y_Q)$ in affine represen-

tation. Let $P = (X_p, Y_p, Z_p)$ denoted by its Jacobian projective representation, where $P = [k]Q$ is computed by the *Left-to-right* algorithm. Let us guess a sequence of bits $k = \{k_{l-1}, \ldots, k_0\}$, starting from its least significant of bits of k (k_0). Let t be a small integer and once that t bits of secret key k are guessed then it is possible to compute a set of candidates for the coordinates of the sequence of intermediate values handled by the *double-and-add* algorithm while k's bits are processing and t bits are tracking (that appear at the end of the algorithm). This is achieved by reversing computations: reversing doubling is halving and reversing an addition amounts to subtracting Q. Thus, we obtain a set of sequences: $\{s_1, s_2, \ldots, s_m\}$ where $s_j = \{M_0^{(j)} \rightarrow M_1^{(j)} \rightarrow \cdots \rightarrow M_l^{(j)}\}$, of the intermediate points, with $M_l^j = P$. Let $M_i = (x_i, y_i)$ points in affine coordinates. The corresponding point in Jacobian projective coordinate is denote by (X_i, Y_i, Z_i). There are two cases: (a) When the step $M_i \rightarrow M_{i+1}$ is an addition the following steps: $Z_{i+1} = (X_i Z_G^2 - X_G Z_i^2) Z_i Z_G = (x_i - x_G) Z_i^3$. Given, $X_i = x_i Z_i^2$. Then $Z_{i+1}/Z_i^3 = (x_i - x_G)$. Here, we need to compute a cubic root to get $Z_i = \sqrt[3]{Z_{i+1}/(x_i - x_G)}$ from Z_{i+1}. (b) When the step $M_i \rightarrow M_{i+1}$ is a doubling the following steps: $Z_{i+1} = 2Y_i Z_i$ which yields $Z_{i+1}/Z_i^4 = 2y_i$. Here, we need to compute a fourth root to get Z_i from Z_{i+1}. Furthermore, in [99] a countermeasures to the *Attack of Projective Coordinates* is presented, Naccache *et al* proposes to replace the output of the computation by $(X, \epsilon Y, \epsilon Z)$ where ϵ is chosen randomly from $\{-1, 1\}$. Indeed, the affine representation of the result is not affected by this modifications as $(X, \epsilon Y, \epsilon Z) \xrightarrow[\text{affine}]{} \left(\frac{X}{(\epsilon Z)^2}, \frac{\epsilon Y}{(\epsilon Z)^3}\right) = \left(\frac{X}{Z^2}, \frac{Y}{Z^3}\right) = (x, y)$. For details see [99].

On the other hand, Smart *et al.* in [122] demonstrated a relationship between use of *Randomized Field Arithmetic* and *Randomized Projective Coordinates*, providing a concrete comparison of their security and performance properties. The authors presented a framework and explained the causes of RPA, since for specific values of P one can produce $R(P)$ ($R(P)$ defined as projective coordinates class of P) whose bits are highly correlated with those of $P = (0, y)$ and show that it is the best countermeasure *redundant modular arithmetic*, see Section 5.10.

5.8.3 Big Mac Attack

This countermeasure does not protect against this attack presented in section 8.

5.8.4 Particular Point Attack

The attack takes advantage of the final conditional reduction of the *Montgomery multiplication Algorithm* (MMA). This attack is feasible only in curves with parameters $a = -3$ (NIST [101]), such curves allow to calculate $3(X_1 + Z_1^2)(X_1 - Z_1^2)$ for doubling algorithm. The attack exploits the occurrence of a special point: $P = (2, y)$. On Jacobian coordinates $P = (2Z_1^2, yZ_1^3, Z_1)$ for some $Z_1 \in \mathbb{F}_p^*$. When P is doubling, its coordinates are replaced by $C = 3(3Z_1^2)(Z_1^2)$. In [123], the authors show that the reduction probability during the MMA of α and β is high, if the relation $\beta = 3\alpha$ is satisfied for the random values. If the attacker carefully chooses the base point, this point occurs only at a certain scalar hypothesis. In this case, the average time of the scalar multiplication is higher than the random inputs. The particular point is not well randomized by the *Random Projective Coordinates Countermeasure*. In fact, whatever the value of Z_1 in the entries of the modular multiplication when C is calculated, it still has $\alpha, \beta = 3\alpha$ for some $\alpha \in \mathbb{F}_p$. The attack can therefore be applied even if this countermeasure is present.

5.9 Ciet and Joye's Method $2P^*$

In [21] the Method $2P^*$ is proposed. This randomization method is applicable to *Left-to-right* algorithm. The idea is randomize $[2]P$ using the *Random Projective Coordinates* algorithm. This enables continuing to use P in affine coordinate. Then, computing the scalar multiplication a mixed coordinates is used (more efficient to use only projective coordinates). This countermeasure does not present security problems.

5.10 Redundant Modular Arithmetic of Smart *et al.*

Smart *et al.* in [122] presented the following idea. *Let m be a modulus with which we wish to perform modular arithmetic. The standard representation is to take $S = \{0, \ldots, m-1\} = \mathbb{Z}/m\mathbb{Z}$. However, we can also hold elements in a redundant form by taking a range $R = \{0, \ldots, C-1\}$, with $C = c \cdot m$, for an integer c co-prime to m and then holding integers modulo m within this range. Such a redundant field representation can create a defense against the attack of Goubin [51].*For more details see [122].

5.11 Joye and Tymen, Randomized field $\mathbb{K}$ Isomorphism [69]

The idea of this countermeasure is to use a representation of random fields definition of elliptic curve, i.e. use a *Randomized Field* through of the isomorphism $\phi : \mathbb{K} \to \mathbb{K}'$. The latter is used to obtain a point $P' = \phi(P)$ of the curve $E' = \phi(E)$, then the scalar multiplication is calculated by: $[k]P = \phi^{-1}([k](\phi(P)))$. The major disadvantage of this countermeasure is that all fields used by the NIST and SEGC standards are defined by Mersenne primes given which the computational efficiency is better in this fields, but using this technique isomorphisms fields the operation losses performance [6].

Security Problems:

5.11.1 RPA and ZPA

See subsections 7.1 and 7.2.

5.12 Randomized $E(\mathbb{K})$ Isomorphism Joye and Tymen [69]

The idea of this countermeasure is to transfer the base point $P_1 = (x, y) \in E_1(\mathbb{K})$ to randomly isomorphic curve $\phi : E_1(\mathbb{K}) \to E_2(\mathbb{K})$ (the parameters of the curve $E_2(\mathbb{K})$ are $a' = r^4 a$ and $b' = r^6 b$, b is not needed in the scalar multiplication algorithm), the transferred point is $\phi(P_1) = (r^2 x, r^3 y) = P_2$ and execute the scalar multiplication ($[k]P_2 = [k]\phi(P_1)$) on the curve $E_2(\mathbb{K})$ and bring the result $Q_2 = (x_k, y_k)$ back to the original curve $E_1(\mathbb{K})$ and we compute $Q_1 = [k]P = (x_k/r^2, y_k/r^3) = \phi^{-1}([k](\phi(P)))$. The randomization takes $4M + 2S$ at the beginning and $1I + 3M + 1S$ at the end. However, when random isomorphisms curve is used the parameters of $E_2(\mathbb{K})$ cannot be chosen and the curve parameter a is randomized, this implies that fast doubling formula for $a = -3$ cannot be used.

Generalization: Tunstall and Joye in [124] define $\phi(P) = P' = (X', Y', Z') = (f^\mu X, f^\nu Y, Z)$ for an arbitrary $f \in \mathbb{F}_p - \{0\}$ and some small integer μ and ν. The inverse of ϕ can be computed without inverting f since $P = \phi^{-1}(P') = (f^\nu X', f^\mu Y', f^{\mu+\nu} Z)$. For the case $\mu = 2$, $\nu = 3$ correspond to the technique of randomized $E(\mathbb{K})$ isomorphism Joye and Tymen [69].

Security Problems:

5.12.1 RPA, ZPA and SVA

See subsections 7.1, 7.2 and 7.3.

5.12.2 Big Mac Attack

This countermeasure does not protect against B*ig Mac* attack presented in 8.

5.13 Summary Performance & Security Problems Countermeasures of Randomization Point in ECC

We can observe that (a) This countermeasure is considered inefficient, since it must perform two scalar multiplications $S = [k]P$ and $[k](P + R)$. (b) This countermeasure has a very low cost since only a few multiplications are required: $3M$ for the homogeneous representation and $4M + 1S$ for the Jacobian representation. (c) Mersenne or "sparse" primes cannot be used. (d) $a = -3$ cannot be used.

Attacks Summary:

(ϕ) Okeya and Sakurai Analysis [102], Fouque's Doubling Attacks [40], Big Mac Attack [127]. (φ) Goubin's Attacks (RPA) [51], Akishita-Takagi Attacks (ZPA) [2], SVA of Murdica [96], Projective Coordinates Leak of Naccache [99], Big Mac Attack [127], Particular Point Attack [123]. (χ) Goubin's Attacks (RPA) [51], Akishita-Takagi Attacks (ZPA) [2], SVA of Murdica [96]. (ψ) Goubin's Attacks (RPA) [51], Akishita-Takagi Attacks (ZPA) [2], SVA of Murdica [96], Big Mac Attack [127].

Table 6: Comparison of the Different DPA Countermeasures

Countermeasure	**Total Cost**	**Security Problem**
Blindig the Point Second Countermeasure of Coron's, [24]	High$^{(a)}$	(ϕ)
Randomized Projective Coord. of Coron's, [24]	Low$^{(b)}$	(φ)
Method $2P^*$ of Ciet and Joye's [21]	Negligible	
Redundant Modular Arithmetic of Smart [122]	Low	
Randomized Field $\mathbb{K}$ Isomorphism of Joye and Tymen [69]	High$^{(c)}$	(χ)
Randomized $E(\mathbb{K})$ Isomorphism of Joye and Tymen [69]	High$^{(d)}$	(ψ)

6 Countermeasure Adress-bit DPA

Itoh *et al.* presented the address-bit DPA (ADPA) [65], it exploits and uses the leaked information guesses individual bits of memory addresses or temporal register. For example, an implementation of Algorithm 3 *Modified-Montgomery-ladder* presented in [74], the address of the doubled point only depends on k_i. As a result, k_i can be recovered if the attacker can distinguish between data read from R_0 and from R_1.

6.1 Itoh *et al.*'s Countermeasure

In order to protect from this attacks Itoh *et al.* [64] presented the Algorithm 11. The registers on steps 6 and 7 are masked with random numbers r.

Algorithm 11 Montgomery powering ladder method with randomized address [64]

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (1, k_{n-2}, \ldots, k_1, k_0)_2$ and $r = (r_{n-1}, \ldots, r_1, r_0)_2 \in \mathbb{N}$, where r_i are random numbers
Outputs: $x([k] \cdot \mathbf{P})$
1: $R_{r_{n-1}} \leftarrow x(\mathbf{P})$;
2: $R_{1-r_{n-1}} \leftarrow 2R_{r_{n-1}}$
3: **for** i **from** $n-2$ **to** 0 **do**
4: $\quad R_2 \leftarrow 2(R_{k_i \oplus r_{i+1}})$
5: $\quad R_1 \leftarrow R_0 + R_1$
6: $\quad R_0 \leftarrow R_{2-(k_i \oplus r_i)}$
7: $\quad R_1 \leftarrow R_{1+(k_i \oplus r_i)}$
8: **end for**
9: **return** R_{r_0}

In the next section we present a security poblem discovered by Izumi.

6.2 Izumi *et al.*'s Countermeasure

Izumi *et al.* in [67] observed, *"When a register is overwritten by the same data as one stored in the register during a data move process, the power consumption is lower than the case of being overwritten by the different data"*. As shown in Step 7 of the Algorithm 11, R_1 stores two different data on the relation between l-th secret key bit k_l and random bit r_l. $R_1 \leftarrow R_1$ if $k_l = r_l$; $R_1 \leftarrow R_0$ if $k_l \neq r_l$. Izumi's *el al.* attack analyzed the following debility: l-th loop and the beginning of the $(l-1)-$th loop the following calculation is performed: l^{th} step 7 : $R_1 \leftarrow R_{1+(k_l \oplus r_l)}; (l-1)^{th}$ step 4 : $R_2 \leftarrow 2(R_{k_{l-1} \oplus r_l})$. In step 7 of the l-th loop, we assume that power traces corresponding to $k_l \oplus r_l = 1$ is in the group $A := \{$*The power traces which are higher than the threshold*$\}$ and power traces which is related to $k_{l-1} \oplus r_l = 0$ is in group $B := \{$*The power traces which are lower than the threshold*$\}$. The address value $k_{l-1} \oplus r_l$ of the source register $R_{k_{l-1} \oplus r_l}$ in the step 4 of the $(l-1)-$th loop is calculated as follows. For the group A, $k_{l-1} \oplus r_l$ becomes $\overline{k_{l-1} \oplus k_l}$ since $r_l = \overline{k_l}$. On the other hand, $k_{l-1} \oplus r_l$ becomes $k_{l-1} \oplus k_l$ since $r_l = k_l$ in the group B. As a result, the random bit r_l can be cancelled out in step 4. Then apply a DPA to calculate the difference between power consumption (P_w) in the group A and the group B during the data move process. $Pw(1 \rightarrow \overline{k_{l-1} \oplus k_l}) - Pw(1 \rightarrow k_{l-1} \oplus k_l)$. Since k_l and k_{l-1} are both constant, we can distinguish whether k_l is equal k_{l-1} or not by ADPA. Izumi, proposes a new algorithm for to resolve this problem, it is showed in the Algorithm 12. This new countermeasure does not present security problems.

7 Countermeasure for RPA, ZPA & SVA

In this section we presents the different countermeasures for RPA, ZPA and SVA. Particulary the SVA is presented in Section 7.3. The attacks RPA, ZPA and SVA assume that the secret scalar k is fixed and the attacker can be chosen the point P in the scalar multiplication, therefore this attack applies to the following protocols ECIES and single pass ECDH but not in ECDSA and two-pass ECDH. To protect against RPA, ZPA and SVA, the base point P or the secret scalar d should be randomized.

Algorithm 12 Montgomery powering ladder method with randomized address [67]

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (1, k_{n-2}, \ldots, k_1, k_0)_2$ and $r = (r_{n-1}, \ldots, r_1, r_0)_2 \in \mathbb{N}$
Outputs: $x([k] \cdot \mathbf{P})$
1: $R_{r_{n-1}} \leftarrow x(2 \cdot \mathbf{P})$;
2: $R_{1 \oplus r_{n-1}} \leftarrow x(P)$
3: **for** i **from** $n-2$ **to** 0 **do**
4: $\quad R_2 \leftarrow 2(R_{k_{i+1} \oplus k_i \oplus r_{i+1}})$
5: $\quad R_{1 \oplus r_i} \leftarrow R_0 + R_1$
6: $\quad R_{r_i} \leftarrow R_2$
7: **end for**
8: **return** $R_{k_0 \oplus r_0}$

7.1 Countermeasures RPA

The idea presented by Goubin in [51] was explained in subsection 4.2.2.

7.1.1 Point Co-factor Countermeasures Special Point Small Order $(x, 0)$

In [112] the next theorem is presented: *"For an elliptic curve E over prime field $\mathbb{K} = \mathbb{F}_p$ $(p > 3)$, a special point of the form $(x, 0)$ exists if and only if the equation $x^3 + ax + b = 0$ has at least one root α in $\mathbb{K}$"*. This special point has a small order, to prevent small subgroup attacks, the most protocol can be reformulated using cofactor multiplication. In [112], to prevent this attacks, at the protocol of Diffie-Hellman cofactor variant, Alice first computes $Q = [h]P$ (where h is a cofactor) and then computes the shared secret via $[d]Q$, if and only if $Q \neq P_\infty$. Goubin's attack then no longer applies since only genuine points in the subgroup of order q are passed into the scalar multiplication algorithm with the secret exponent d. We note that elliptic curves defined for standard NIST prime fields considers the cofactor $h = 1$, then do not have special points of low order.

7.1.2 Isogeny Countermeasure for Special Point $(0, y)$

Smart in [112] the next theorem: *"For an elliptic curve $E : y^2 = x^3 + ax + b$ over a prime field $\mathbb{F}_p$ with $p > 3$, a special point of the form $(0, y)$ exists if and only if b is a quadratic residue modulo p, i.e. $\left(\frac{b}{p}\right) = 1$, where $\left(\frac{\cdot}{\cdot}\right)$ is the Legendre symbol"*. In order to resist RPA for point of large order, Smart in [112] proposed to map the underlying curve to the isogenous curve that does not have the point $(0, y)$. This countermeasure with a small isogeny degree is faster. However, but it is much slower to implement it on a micro-controller, for more details see [6].

7.1.3 Volcanoes Isogeny

Miret *et al.* in [90], presented improvements to the searching time to find the isogeny of SECG elliptic curves [114], for the above the Isogeny-Volcano is used. In Table 7 we present the result obtained by Smart in [112] and Mired *et al.* in [90]. In the second and fourth column the minimal and preferred isogeny degree with respect to condition ED4 is presented by Smart in [112], while the third and the fifth columns contain the degrees of the isogeny-route given by the algorithm presented by Miret in [90]. More precisely, the minimal $\ell_{\mathbf{std}}$ and preferred isogeny degree $\ell_{\mathbf{prf}}$ can be defined by: ℓ_{std} The minimal Isogeny degree for condition ED4. ℓ_{prf} The minimal Isogeny degree for condition ED4 and condition $a = -3$. The integer $\ell_{\mathbf{std}} - route$ and $\ell_{\mathbf{prf}} - route$

correspond to the minimal and preferred isogeny degree obtained by Miret *et al.* in [90]. For example, the curve P-192 the preferred isogeny degree is $\ell_{\mathbf{prf}} = 73$ while $\ell_{\mathbf{prf}} - route = 5-13-23$, which means that three isogenies of degrees 5, 13 and 23 are compound.

Table 7: Minimal isogeny degrees with respect to ED4 for SECG curves

ED4	$\ell_{\mathbf{std}}$ [112]	$\ell_{\mathbf{std}} - route$ [90]	$\ell_{\mathbf{prf}}$[112]	$\ell_{\mathbf{prf}}$-route [90]
P-192	23	5-13	73	5-13-23
P-224	1	1	1	1
P-256	3	3	11	3-5
P-384	19	19	19	19
P-521	5	5	5	5

7.1.4 Isomorphism Shifting

In [34] a countermeasure against RPA is presented in order to avoid the points of the form $(0, y)$. The idea is try to control the point using an isomorphism, given the next definition of the elliptic curve: $E : y^2 = x^3 + a_4x + a_6, \quad E' : y^2 = x^3 + a'_2x^2 + a'_4x + a'_6$, are isomorphic over $\mathbb{F}_p$ if only if there exist $u \in \mathbb{F}_p^*$ and $r \in \mathbb{F}_p$ such that the change of $(x, y) \to (u^{-2}(x - r), u^{-3}y)$, transforms equation E into equation E' with: $u^2a'_2 = 3r,\ u^4a'_4 = a_4 + 3r^2,\ u^6a'_6 = a_6 + ra_4 + r^3$. The computational costs is $2M + 1S + 5A$. For more details see [34].

7.2 Countermeasure ZPA

The ZPA was presented in subsection 4.2.3. Akishita *et al.* in [3] presented the following proposition to know when an elliptic curve has points satisfying ED1: $3x^2 + a = 0$. *Let E be an elliptic curve over prime field $\mathbb{F}_p$ defined by $y^2 = x^3 + ax + b$. The elliptic curve E satisfies $a = -3$, $\#E(\mathbb{F}_p)$ is odd, and $\left(\frac{a}{p}\right) = -1$. Then the elliptic curve E satisfies condition ED1.*

The Table 8 shows the comparison of the results obtained by Akishita-Takagi and Miret *et al.* presented in [90], concerning isogenous curve and conditions ED1 and ED4 with results obtained using the *Isogeny-route*.

Table 8: Minimal isogeny degrees with respect to ED1+ED4 for SECG curves

ED1+ED4	$\ell_{\mathbf{std}}$	$\ell_{\mathbf{std}} - route$	$\ell_{\mathbf{prf}}$	$\ell_{\mathbf{prf}}$-route
P-192	23	13-13	–	–
P-224	1	1	1	1
P-256	3	3	23	5-11
P-384	31	31	–	–
P-521	5	5	5	5

7.3 Countermeasure SVA

Murdica *et al.* in [96] presented an attack called *Same Value Analysis*. The attacker, to observe at points that show up same values during *addition or doubling* algorithm, uses an internal collisions power analysis attack to detect if the special point appears during an scalar multiplication. Murdica *et al.*, notes that certain special points, internal collisions occur in the operation

of doubling Jacobian coordinates, for example, SED2: $x = 1$, SED3: $y = x^2$ and SED15: $2y = 3x^2 + a$, more specifically. Let $P = (X_1, Y_1, Z_1) = (\lambda_1^2 x_1, \lambda_1^3 y_1, \lambda_1)$ be a point in Jacobian coordinates one can be computed the doubling point by the following formula $P_3 = (X_3, Y_3, Z_3)$: $\alpha = 3X_1^2 + aZ_1^4$, $\beta = 4X_1Y_1^2$, $Z_3 = 2Y_1Z_1$, $X_3 = \alpha^2 - 2\beta$, $Y_3 = \alpha(\beta - X_3) - 8Y_1^4$. For example, when SED2 is used, the latter condition implies that during the computation of the X_1^2 and Z_1^4 the energy consumption are the same. He uses the methodology presented by [111] and [28] to detect internal collision. If a collision is detected, he can conclude that $k_i = 0$. Otherwise, he concludes that $k_i = 1$. The attacker can recursively recover all bits of the private key k.

In Table 9, we can see there are no NIST curves that are safe for RPA, ZPA or SVP.

Table 9: Summary of RPA, ZPA and SVP Points on Standard Curves NIST [101]

$E : y^2 = x^3 - 3x + b$						
Curves $\mathbb{F}_p$	RPA	ZPA		SVA		
	$(0, y)$	$3x^2 + a = 0$	$5x^4 + 2ax^2 - 4bx + a^2 = 0$	$x = 1$	$x^2 = y$	$2y = 3x^2 + a$
P-192	✓	✓	✓	ø	ø	✓
P-224	ø	ø	✓	ø	ø	✓
P-256	✓	ø	✓	ø	ø	✓
P-384	✓	✓	ø	ø	ø	ø
P-521	✓	ø	ø	✓	✓	ø

8 Countermeasures against Big Mac Attack

The scalar multiplication implicitly must operate with modular long integer multiplication, there are several methods to perform multiplication, but the most used is the schoolbook long integer multiplication that uses a t-bit internal multiplier giving a $2t$-bit result. For more detail see [56] Chapter 2. *Big Mac attack* was introduced by Walter [127] and basically consists of detecting if two multiplications share a common operator by comparing their energy traces. The success of the attack depends directly on the length of the integers to be used. This method was generalized by Bauer in [17] for atomic blocks. Recently Danger in [32] improves the attack presented in [17] since it is able to compare many multiplications, in particular he used 14 pairs instead of two.

8.1 Multiplication with Random Permutation

This countermeasure was introduced by Clavier [27] and consists of randomizing the order of the manipulation of the words x_i during a long multiplication. Here, the correlation between the common operators is hidden. The number of possibilities of internal multiplication is $(l!)^2$, for more details see [27]. This countermeasure also protects against Horizontal Correlation Analysis on Exponentiation [9].

Recently, Bauer [17] shows that the Big Mac attack is still feasible againts Clavier coutermeasure [27], since it is still possible to perform this attack by deducting only one of the random permutations instead of both. With this attack, the number of possibilities is reduced to $l!$ for $l \leq 16$. The authors in [17] suggest the countermeasure *generation random permutation* over

$\{(a,b); a,b \in [0;l]\}$ and this ideas is presented in Algorithm 13. The output of the above algorithm is used to randomize the manipulation of the words $X[a]$ and $Y[b]$ simultaneously. Here, a second permutation P must be developed in order to avoid attacks on the *carry propagation* treatment on the integers $1, 2, \ldots, 2l+1$, as observed in the Algorithm 14. This countermeasure (Algorithm 14) does not have security problems (For more details see [17]).

Algorithm 13 Generation of Random Permutation (GRP)

Inputs: Two intergers t and l, a permutation α_0 over $[0, (t+1)^2 - 1]$.
Outputs: A vector in $[0, (t+1)^2 - 1]$ (elements are represented in base $t+1$)

$(r_0, r_1, \ldots, r_{l-1}) \leftarrow$ random elements in $\mathbb{Z}_{(t+1)^2}$
for i **from** 0 **to** $l-1$ **do**
  for j **from** 0 **to** $(t+1)^2 - 1$ **do**
    $\alpha_{i+1}[j] \leftarrow \alpha_0[(\alpha_i[j] + r_i) \bmod (t+1)^2]$
  end for
end for
return α_l

Algorithm 14 Long Interger Multiplication with randomization of the two loops together.

Inputs: $X = (X[t], X[t-1], \ldots, X[0])_{2^w}, Y = (Y[t], Y[t-1], \ldots, Y[0])_{2^w}$, p.
Outputs: LIM(X, Y).

$\alpha_l = (\alpha, \beta) \leftarrow \mathrm{GRP}(t, p, \alpha_0)$
$P \leftarrow$ random permutation of $1, 2, \ldots, 2t+1$.
for a **from** 0 **to** $2t+1$ **do**
  $R[a] = C[a] = 0$
end for
for h **from** 0 **to** $(t+1)^2 - 1$ **do**
  $a \leftarrow \alpha[h]$; $b \leftarrow \beta[h]$
  $(U, V)_{2^w} \leftarrow R[a+b] + X[a] \cdot Y[b]$
  $R[a+b] \leftarrow V$
  $C[a+b+1] \leftarrow C[a+b+1] + U$
end for
for i **from** 1 **to** $2t+1$ **do**
  for j **from** 1 **to** $2t+1$ **do**
    $s \leftarrow P[j]$
    if $s \geq i$ **then**
      $(U, V)_{2^w} \leftarrow R[s] + C[s]$
      $R[s] \leftarrow V$
      $C[s+1] \leftarrow C[s+1] + U$
      $C[s] \leftarrow 0$
    end if
  end for
end for
return R

9 Countermeasures for Several Attacks

In this section, countermeasures that protect against several simultaneous attacks were analyzed.

9.1 BRIP Countermeasure of Mamiya *et al.*'s

Mamiya *et al.* presented a countermeasure for attacks SPA, DPA, RPA, ZPA and SVA called BRIP, the latter, only work for *Left-to-right* algorithm presented in [93]. This method uses a

random initial point R (thus, it is resistant against DPA, RPA, ZPA and SVA), furthermore in order for this algorithm to be secure against SPA, it is based on the principle of *Doubling-and-add always* of Coron's. This algorithm computes $[k]P+R$ and at the end of the algorithm execution computes $([k]P+R)-R=[k]P$. Mamiya *et al.* applies the identity $1=(1\overline{11}\cdots\overline{11})_2$ apply the extended binary method [75] to compute: $[k]P+R=[(d_{n-1}d_{n-2}\cdots d_1d_0)_2]P+[(1\overline{11}\cdots\overline{11})_2]R$. R is subtracted at the end to the algorithm execution. The Algorithm 15 cost per bit using General Jacobian coordinates is $15M+9S$, if $a=-3$ then the cost is $15M+7S$.

Algorithm 15 BRIP

Inputs: Point $\mathbf{P}\in E(\mathbb{F}_q)$, $k=(k_{n-1},\ldots,k_1,k_0)_2\in\mathbb{N}$
Outputs: $\mathbf{Q}=[k]\cdot\mathbf{P}$
1: $R\leftarrow$ *randompoint*(); $R_0\leftarrow R$; $R_1\leftarrow -R$; $R_2\leftarrow P-R$
2: **for** i **from** $n-1$ **to** 0 **do**
3: $R_0\leftarrow 2R_0$
4: **if** $k_i=0$ **then**
5: $R_0\leftarrow R_0+R_1$
6: **else**
7: $R_0\leftarrow R_0+R_2$
8: **end if**
9: **end for**
10: **return** R_0+R_1

Security Problems:

2-torsion Attacks Yen *et al.* in [129] presented a chosen-message-attack for the RSA. Here, an attacker by just analyzing a single power consumption curve can know k_i bits of the secret key, it is only applicable to ECC defined in fields of characteristic two. Besides, it can be easily avoided: before computing $[k]P$ it is mandatory check $2P\neq P_\infty$, for more details see [92].

Doubling attacks BRIP can be attacked through *doubling attack*. The reason is that the intermediate values of BRIP are always of the form $X+R$, where R is Random Initial Point (RIP) and X is the original unmasked intermediate values in each scalar multiplication, for more details see [53].

Address-bit DPA See Section 6.

9.2 Kim *et al.*'s Countermeasure

Kim *et al.* in [79] presented an countermeasure against the DPA, RPA, ZPA, SVA, Doubling Attack, and 2-torsion attacks. This is based on random blinding point countermeasure ($P+R$ with R a random point) with the *Shamir-Strauss method*. The basic idea of this countermeasure is to compute: $[k]P=[k](P+R)+[\#E-k]R=\sum_i\{k_i(P+R)+s_iR\}=\sum_i k_iP+\sum_i(k_i+s_i)R=[k]P+(k+r)R=[k]P+(\#E)R$ where $(\#E)R=P_\infty$ as described in Algorithm 16. The computational cost is $2A+1D$ for precalculation and $nD+nA$ for the computation of scalar multiplication $[k]P$. For this algorithm the next attack can be performed, in the case that $k_is_i=00$, it must be added in line 6 of the algorithm $Q=Q+T_{k_is_i}=Q+T_{00}=Q+P_\infty$, in this particular case a SPA attack could be apply.

Algorithm 16 Kim's Countermeasure

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Outputs: $\mathbf{Q} = [k] \cdot \mathbf{P}$
1: $s = \#E - k$; Choose a random elliptic point $R \in E(\mathbb{F}_q)$
2: $T_{00} \leftarrow P_\infty$; $T_{01} \leftarrow R$; $T_{10} \leftarrow P + R$; $T_{11} \leftarrow P + 2R$
3: $Q \leftarrow T_{00}$
4: **for** i **from** $n - 1$ **to** 0 **do**
5: $\quad Q \leftarrow 2Q$
6: $\quad Q \leftarrow Q + T_{k_i s_i}$
7: **end for**
8: **return** Q

9.3 Ha *et al.*'s Countermeasure

Ha *et al.* in [53], presented an enhanced countermeasure using the Shamir's trick and a *message blinding technique*. The proposed countermeasure can protect against SPA, DPA, Doubling attacks, RPA, ZPA, SVA, 2-torsion attacks and address-bit DPA. The basic idea of the proposed countermeasure is blind a point P using a random point R. Here, it is assumed that the number of points on the curve E represented by $\#E$ is the large $n-$bits. Thus, $t(P + R) + sR + (2^n - 1)(P+R)$ is finally computed instead of $[k]P$, where t and s are $n-$bit positive integers. The final result $[k]P$ is obtained by: $[k]P = (d\#E+k-(2^n-1))(P+R)+(\#E-k)R+(2^n-1)(P+R) = \sum_i^{n-1} 2^i(t_i(P + R)) + s_iR + (P + R))$, where $\#ER = P_\infty$. Let $t = d\#E + k - (2^n - 1)$ and $s = \#E - k$ be n-bit integers, then the smallest integer d is chosen such that $(d - 1)\#E + k < (2^n - 1) < d\#E + k$, thus d is 1 or 2. The idea of the algorithm is to simultaneously compute the above three operations $t(P + R)$, sR, and $(2^n - 1)(P + R)$, as is described in Algorithm 17. The computational cost is $4A$ for pre-calculation and $(n - 1)D + (n - 1)A$ for the computation of scalar multiplication $[k]P$. This algorithm has not been attacked.

Algorithm 17 Ha's Countermeasure

Inputs: Point $\mathbf{P} \in E(\mathbb{F}_q)$, $k = (k_{n-1}, \ldots, k_1, k_0)_2 \in \mathbb{N}$
Outputs: $\mathbf{Q} = [k] \cdot \mathbf{P}$
1: $t = d\#E + k - (2^n - 1)$; $s = \#E - d$
2: Choose a random Elliptic point R and random bits u, v
3: $T_{00\oplus uv} \leftarrow P + R$; $T_{01\oplus uv} \leftarrow P + 2R$; $T_{10\oplus uv} \leftarrow 2P + 2R$; $T_{11\oplus uv} \leftarrow 2P + 3R$
4: $Q \leftarrow T_{t_{n-1}s_{n-1}\oplus uv}$
5: **for** i **from** $n - 2$ **to** 0 **do**
6: $\quad Q \leftarrow 2Q$
7: $\quad Q \leftarrow Q + T_{t_i s_i \oplus uv}$
8: **end for**
9: **return** Q

9.4 Summary Performance v/s Security Problems Countermeasures for Several Attacks in ECC

We will consider the cost of generating a random point R on the curve as R_{ran}.

Algorithm	# regs.	Total cost	Security Problem
BRIP Algorithm [93]	3	$(n + 3)A + nD + R_{ran}$	τ
Kim's Algorithm [79]	4	$(n + 2)A + (n + 1)D + R_{ran}$	υ
Ha's Algorithm [53]	5	$(n + 3)A + (n - 1)D + R_{ran}$	

Attacks: (τ): 2-torsion Attacks, Address-bit DPA and Doubling attacks. (υ): SPA Attack when $k_i s_i = 00$.

10 Summary

The Table 10 shows a summary of attacks versus countermeasures. Table 11 shows a summary of the side channel attacks versus the main features for the implementation of the different attacks.

Table 10: Summary countermeasures & security problems

Attacks	Countermeasures	Security Problems (Attacks)
Simple Side Channel Attacks	Unified Formulae	Izu-Takagi, Walter's, Amiel's, PACA, Horizontal Collision Correlation Analysis, Horizontal SVA,
	Double-and-Add-Always	Doubling Attack, RPA, ZPA, C-safe Fault Attack, M-safe Fault Attack, 2-Torsion Attack, Correlation Collision Attack on horizontal settings
	Montgomery Ladder	Twisted Curve Fault Attacks x-only version, RPA, ZPA, Relative Doubling Attack, Address-bit DPA, Correlation Collision Attack on horizontal settings, Unsupervised learning Attack, Cmov Attack
	Joye's Double-Add	
	Joye's Add-only	Correlation Collision Attack on horizontal settings
	Zero-less Signed-Digit	
	Atomic Blocks	Horizontal Collision Correlation, Vertical Collision Correlation, Doubling Attack, Chen's Attack
Differential Side Channel Attacks	Coron's First Countermeasure	Okeya-Sakurai Analysis, Doubling Attacks, Ciet-Joye Analysis, Fouque's Carry Leakage, Feix's Attacks, Big Mac Attack
	Clavier-Joye	Ebeid Analysis, Muller-Valette Attack, Fouque's Carry Leakage Attack, Ha's Analysis using 2-torsion Attack's, Big Mac Attack
	Trichina-Belleza	
	Ciet-Joye	Ebeid Analysis, Ha's Analysis using 2-torsion Attack's
	Self-Randomized Exp.	
	Blinding the Point	Okeya-Sakurai Analysis, Doubling Attack, Big Mac Attack
	Randomized Projective Coord.	RPA, ZPA, SVA, Naccache's Projective Coord. Leak, Big Mac Attack, Particular Point Attack.
	Ciet-Joye Method $2P^*$	
	Smart's Redundant Modular Aritmetic	
	Randomized Field $\mathbb{K}$ Isomorphism	RPA, ZPA
	Randomized $E(\mathbb{K})$ Isomorphic	RPA, ZPA, SVA, Big Mac Attack
Adress-Bit DPA	Itoh's Countermeasure	Izumi ATable 11 shows a summary of the side channel attacks versus the main features for the implementation of the different attacks.ttacks.
	Izumi's Countermeasure	
RPA, ZPA, SVP	Isogeny Countermeasure	
	Volcano Isogeny	
	Isomorphic Shifting	
Big Mac Attack	Multiplication with Random Permutation	
Countermeasures for Several Attacks	BRIP (SPA, DPA, RPA, ZPA)	Doubling Attacks, Address-bit DPA, 2-torsion Attacks
	Kim's (DPA, RPA, ZPA, Doubling Attack, 2-torsion Attack)	SPA
	Ha's (SPA, DPA, Doubling Attacks, RPA, Address-bit DPA, 2-torsion and ZPA)	

11 Conclusion

In this article we give an overview of the countermeasure for passive SCA in ECC, we analyzed their security problems and computational performances for the current countermeasures.

We believe that practical and theoretical analysis of these countermeasures is important for security of ECC in IoT devices. Moreover, we should assume that using some countermeasure

Table 11: Physical Attacks on ECC

Attack	Single Exe.	Multiple Exe.	Chosen Base Point	Using Output Point	Incremental key Recovery
Izu-Takagi [66]		✓	✓	✓	✓
Walter's [134]	✓				
Amiel *et al*'s [4]	✓				
PACA [5]	✓				
Horizontal collision correlation analysis [9]	✓				
Horizontal SVA [34]	✓				
Doubling Attacks [40]		✓	✓		
C-safe Fault [130]		✓		✓	✓
M-safe Fault [131]		✓		✓	✓
The 2-Torsion(only fields char. two) [129]	✓		✓		
RPA [51]		✓	✓		✓
ZPA [2]		✓	✓		✓
Relative Doubling Attacks [128]		✓	✓		✓
Address-bit DPA [64]		✓			✓
Correlation Collision Attack in the Horizontal Setting [57]	✓				
Vertical Collision Correlation SVA [97]		✓			✓
Okeya and Sakurai Analysis [102]		✓			
Fouque's Doubling Attacks [40]		✓			
Ciet and Joye Analysis [21]		✓			
Fouque's Carry Leakage [46]		✓			
Ebeid Analysis DPA [36]		✓			✓
Muller-Vallete Attack [91]		✓ [4]			
Big Mac Attack [127]	✓				✓
Naccache Projective Coordinate Leak [99]				✓	✓
Particular Point Attack [123]		✓	✓		✓
Izumi Attack [67]		✓			
SVA Attack [96]		✓	✓		✓
Unsupervised Learning Attack [58]	✓				
Cmov Side Channel Attack [100]	✓				

may be effective against several attacks, but a full analysis of the software/hardware solutions is required in order to avoid known attacks or introduce new attacks.

Acknowledgements

The authors are grateful for the financial support given by Universidad de Santiago de Chile, USACH, through the Project DICYT Asociativo 061513VC-DAS, Project DICYT 061433AO and Basal Project USA1555.

References

[1] Abarzúa, R., Thériault, N.: Complete atomic blocks for elliptic curves in Jacobian coordinates over prime fields. Latincrypt 2012, LNCS 7533, Springer, pp. 7–55, (2012).

[2] Akishita, T., Takagi, T.: Zero-value point attacks on elliptic curve cryptosystem. ISC 2003, LNCS 2851, Springer, pp. 218–233, (2003).

[3] Akishita, T., Takagi, T.: On the optimal parameter choice for elliptic curve cryptosystems using isogeny. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. 1, pp. 140–146, (2005).

[4] Amiel, F., Feix, B., Tunstall, M., Whelan, C., Marnane, W. P.: Distinguishing multiplications from squaring operations. SAC 2008, LNCS 5381, Springer, pp. 346–360, (2008). https://doi.org/10.1007/978-3-642-04159-42_2

[5] Amiel, F., Villegas, K., Feix, B., Marcel, L.: Passive and active combined attacks. FDTC 2007, IEEE, pp. 92–99, (2007).

[6] Avanzi, R.: Side channel attacks on implementations of curve-based cryptographic primitives. https://eprint.iacr.org/2005/017.pdf, IACR Cryptology ePrint Archive pp. 1–27, (2005).

[7] Bahl, V., Chancre, V., Dungeon, J.: SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 Ad-Hoc wireless networks. MobiCom'04, ACM, New York, NY, pp. 112–117, (2004).

[8] Bajard, J-C., Imbert, L., Liardet, P-Y., Teglia, Y.: Leak resistant arithmetic. CHES 2004, LNCS 3156, Springer, pp. 62–75, (2004).

[9] Bauer, A., Jaulmes, E., Prouff, E., Reinhard, J. R., Wild, J.: Horizontal collision correlation attack on elliptic curves:–Extended Version–. Cryptography and Communications, Vol.7, (1), pp. 91–119, (2014).

[10] Brier, É., Clavier, Ch., Olivier, F.: Correlation power analysis with a leakage model. CHES 2004, LNCS 3156, Springer, pp. 16–29, (2004).

[11] Brier, É., Dèchéne, I., Joye, M.: Unified point addition formulae for elliptic curve cryptosystems. Nova Science Publishers, chapter XIV, pp. 247–256, (2004).

[12] Bernstein, D.J., Lange, T.: Faster addition and doubling on elliptic curves. ASIACRYPT 2007, LNCS 4833, Springer, pp. 29–50, (2007).

[13] Bernstein, D. J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards Curves. AFRICACRYPT 2008, LNCS 5023, Springer, pp. 389–405, (2008).

[14] Bernstein, D.J., Lange, T.: Inverted Edwards Coordinates. AAECC 2007, LNCS 4851, Springer, pp. 20–27, (2007).

[15] Bernstein, D. J., Lange, T.: Explicit-formula database. http://www.hyperelliptic.org/EFD/, (2017).

[16] Billet, O., Joye, M.: The Jacobi model of an elliptic curve and side-channel analysis. AAECC 2003, LNCS 2643, Springer, pp. 34–42, (2003).

[17] Bauer, A., Jaulmes, E., Prouff, E., Wild, J.: Horizontal and vertical side-channel attacks against secure RSA implementations. CT-RSA 2013, LNCS 7779, Springer, pp. 1–17, (2013).

[18] Brier, É., Joye, M.: Weierstraß elliptic curves and side-channel attacks. PKC 2002, LNCS 2274, Springer, pp. 335–345, (2002).

[19] Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. CHES 2002, LNCS 2523, Springer, pp. 13–28, (2003).

[20] Chevallier-Mames, B., Ciet, M., Joye, M.: Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity. IEEE Transactions on Computers, vol. 53, 6, pp. 760–768, (2004).

[21] Ciet, M., Joye, M.: (Virtually) Free randomization techniques for elliptic curve cryptography. ICICS 2003, LNCS 2836, Springer-Verlag, pp. 348–359, (2003).

[22] Clavier, C., Joye, M.: Universal exponentation algorithm. CHES 2001, LNCS 2162, Springer-Verlag, pp. 300–308, (2001).

[23] Chevallier-Mames, B.: Self-randomized exponentiation algorithms. CT-RSA 2004, LNCS 2964, Springer-Verlag, pp. 236–249, (2004).

[24] Coron, J-S.: Resistance against differential power analysis for elliptic curve cryptosystems. CHES 1999, LNCS 1717, Springer, pp. 292–302, (1999).

[25] Cormen, T. H., Leiserson, Ch. E., Rivest, R. L., Stein, C.: Introduction to algorithms. Third Edition, The MIT Press, (2009).

[26] Chen, T., Li, H., Wu, K., Yu, F.: Countermeasure of ECC against side-channel attacks: Balanced point addition and point doubling operation procedure. APCIP 2009, IEEE, pp. 465–469, (2009).

[27] Clavier, Ch., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Horizontal correlation analysis on exponentiation. ICICS 2010, LNCS 6476, Springer, pp. 46–61, (2010).

[28] Clavier, Ch., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Improved collision-correlation power analysis on first order protected AES. CHES 2011, LNCS 6917, Springer, pp. 49–62, (2011).

[29] Cohen, H., Frey, G., Avanzi, R., Doche, Ch., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of elliptic and hyperelliptic curve cryptography. Chapman and Hall/CRC, (2005).

[30] Chmielewski, Ł., Massolino, P.M.C., Vliegen, J., Batina, L., Mentens, N.: Completing the complete ECC formulae with countermeasures. Journal of Low Power Electronics and Applications, vol. 7, 1, (2017).

[31] Giry, D.: Bluekrypt, Cryptographic key length recommendation. https://www.keylength.com/, (2017). Accessed 28 June 2018

[32] Danger, J-L., Guilley, S., Hoogvorst, Ph., Murdica, C., Naccache, D.: Improving the big mac attack on elliptic curve cryptography. The New Codebreakers, LNCS 9100, Springer, pp. 374–386, (2016).

[33] Das, P., Roy, D. B., Boyapally, H., Mukhopadhyay, D.: Inner collisions in ECC: Vulnerabilities of complete addition formulas for NIST curves. AsianHOST 2016, IEEE, pp. 1–6, (2017).

[34] Danger, J. L., Guilley, S., Hoogvorst, Ph., Murdica, C., Naccache, D.: A synthesis of side-channel attacks on elliptic curve cryptography in smart-cards. Journal of Cryptographic Engineering, vol. 3, 4, pp. 241–265, (2013).

[35] Duquesne, S.: Improving the arithmetic of elliptic curves in the Jacobi model. Information Processing Letters, vol. 104, 3, Elsevier, pp. 101–105. (2007).

[36] Ebeid, N. M.: Key randomization countermeasures to power analysis attacks on elliptic curve cryptosystems. University of Waterloo, Phd.D. Electrical and Computer Engineering, (2007).

[37] Edwards, H. M.: A normal form for elliptic curves. Bulletin of the American Mathematical Society, vol. 44, 3, pp. 393–422, (2007).

[38] Francq, J.: Conception et sécurisation d'unités arithmétiques hautes performances pour courbes elliptiques. Université Montpellier II, Ph.D thesis, Informatique, (2010).

[39] Fan, J., Verbauwhede, I.: An updated survey on secure ECC implementations: Attacks, countermeasures and cost. Cryptography and Security: From Theory to Applications, LNCS 6805, Springer, 265–282, (2012).

[40] Fouque, P-A., Valette, F.: The doubling attack why upwards is better than downwards. CHES 2003, LNCS 2779, Springer, pp. 269–280, (2003).

[41] Fan, J., Guo, X., Mulder, E. D., Schaumont, P., Preneel, B., Verbauwhede, I.: State of the art of secure ECC implementations: A survey on known side-channel attacks and countermeasures. HOST 2010, IEEE, pp. 76–87, (2010).

[42] Farashahi, R., Joye, M.: Efficient arithmetic on Hessian curves. PKC 2010, LNCS 6056, Springer-Verlag, pp. 243–260, (2010).

[43] Feix, B., Roussellet, M., and Venelli, A.: Side-Channel analysis on blinded regular scalar multiplications. INDOCRYPT 2014, LNCS 8885, Springer-Verlag, pp. 3–20, (2014).

[44] Fischer, W., Giraud, C., Knudsen, E. W., Seifert, J.P.: Parallel scalar multiplication on general elliptic curves over $\mathbb{F}_p$ hedged against non-differential side-channel attacks. https://eprint.iacr.org/2002/007.ps, Cryptology ePrint Archive, Report 2002/007, (2002).

[45] Fouque, P. A., Lercier, R., Réal, D., Valette, F.: Fault attack on elliptic curve with Montgomery ladder implementation. FDTC 2008, IEEE, pp. 92–98, (2008).

[46] Fouque, P. A., Réal, D., Valette, F., Drissi, M.: The carry leakage on the randomized exponent countermeasure. CHES 2009, LNCS 5154, Springer-Verlag, pp. 198–213, (2008).

[47] Giraud, Ch., Verneuil, V.: Atomicity improvement for elliptic curve scalar multiplication. CARDIS 2010, LNCS 6035, Springer-Verlag, pp. 80–101, (2010).

[48] Gandolfi, K., Mourtel, Ch., Olivier, F.: Electronic analysis: Concrete results. CHES 2001, LNCS 2162, Springer, pp. 251–261, (2001).

[49] Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual information analysis. CHES 2008, LNCS 5154, Springer, 426–442, (2008).

[50] Giry, D., Quinsquater, J. J.: Bluekrypt cryptographic key length recommendation. http://www.keylength.com/., accessed on 2017. (2011).

[51] Goubin, L.: A refined power-analysis attack on elliptic curve cryptosystems. PKC 2003, LNCS 2567, Springer, 199–211, (2002).

[52] Goundar, R. R., Joye, M., Miyaji, A., Rivain, M., Venelli, A.: Scalar multiplication on Weierstraß elliptic curves from Co-Z arithmetic. Journal of Cryptographic Engineering, vol. 1, (2), pp. 161–176, (2011).

[53] Ha, J., Park, J., Moon, S., Yen, S.: Provably secure countermeasure resistant to several types if power attack for ECC. WISA 2007, LNCS 4867, Springer-Verlag, pp. 333–344, (2007).

[54] Hanley, N., Tunstall, M., Marnane, W. P.: Using templates to distinguish multiplications from squaring operations. International Journal of Information Security, vol. 10, (4), pp. 255–266, (2011).

[55] Hisil, H., Carter, G., Dawson, E.: New formulae for efficient elliptic curve arithmetic. INDOCRYPT 2007, LNCS 4859, Springer, pp. 138–151, (2007).

[56] Hankerson, D., Menezes, A., Vanstone, S.: Guide to elliptic curve cryptography. Springer-Verlag, (2004).

[57] Hanley, N., Kim, H. S., Tunstall, M.: Exploiting collisions in addition chain-based exponentiation algorithms using a single trace. CT-RSA 2015, LNCS 9048, pp. 431–448, Springer, (2015).

[58] Heyszl, J., Ibing, A., Mangard, S., De Santis, F., Sigl, G.: Clustering algorithms for non-profiled single-execution attacks on exponentiations. CARDIS 2013, LNCS 8419, pp. 79–93, Springer, (2014).

[59] Heyszl, J., Mangard, S., Heinz, B., Stumpf, F., Sigl, G.: Localized electromagnetic analysis of cryptographic implementations. CT-RSA 2012, LNCS 7178, Springer, pp. 231–244, (2012).

[60] Heyszl, J., Merli, D., Heinz, B., De Santis, F., Sigl, G.: Strengths and limitations of high-resolution electromagnetic field measurements for side-channel analysis. CARDIS 2012, LNCS 7771, Springer, pp. 248–262, (2013).

[61] Hisil, H., Wong, K. K-H., Carter, G., Dawson, E.: Twisted Edwards curves revisited, ASIACRYPT 2008, LNCS 5350. Springer, pp. 326–343, (2008).

[62] Hisil, H., Wong, K. K-H., Carter, G., Dawson, E.: Jacobi quartic curves revisited. ACISP 2009, LNCS 5594, Springer, pp. 452–468, (2009).

[63] Izu, T., Takagi, T.: A fast parallel elliptic curve multiplication resistant against side channel attacks. PKC 2002, LNCS 2274, Springer, pp. 280–296 (2002).

[64] Itoh, K., Izu, T., Takenaka, M.: A practical countermeasure against address-bit differential power analysis. CHES 2003, LNCS 2779, Springer, pp. 382–396, (2003).

[65] Itoh, K., Izu, T., Takenaka, M.: Address-bit differential power analysis of cryptographic schemes OK-ECDH and OK-ECDSA. CHES 2002, LNCS 2523, Springer, pp. 129–143, (2003).

[66] Izu, T., Takagi, T.: Exceptional procedure attack on elliptic curve cryptosystems. PKC 2003, LNCS 2567, Springer, pp. 224–239, (2003).

[67] Izumi, M., Ikegami, J., Sakiyama, K., Ohta, K.: Improved countermeasure against Addres-bit DPA for ECC scalar multiplication. DATE 2010, IEEE, pp. 981–984, (2010).

[68] Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., Qiu, D.: Security of the internet of things: Perspectives and challenges. Wireless Networks, Vol. 20, (8), pp. 2481–2501, (2014).

[69] Joye, M., Tymen, Ch.: Protections against differential analysis for elliptic curve cryptography. CHES 2001, LNCS 2162, Springer pp. 377-390, (2001).

[70] Joye, M., Tustall, M.: Fault analysis in cryptography. Information Security and Cryptography, Springer, (2012).

[71] Joye, M.: Fast point multiplication on elliptic curves without precomputation. WAIFI 2008, LNCS 5130, Springer, pp. 36–46 (2008),

[72] Joye, M., Tibouchi, M., Vergnaud, D.: Huff's model for elliptic curves. ANTS 2010, LNCS 6197, Springer, pp. 234–250, (2010).

[73] Joye, M.: Highly regular right-to-left algorithms for scalar multiplication. CHES 2007, LNCS 4727, Springer, pp. 135–147, (2007).

[74] Joye, M., Yen, S-M.: The Montgomery powering ladder. CHES 2002, LNCS 2523, Springer, pp. 291–302, (2003).

[75] Knuth, D. E.: The art of computer programming. Vol, 2: Seminumerical algorithms. Addison-Wesley, (1981).

[76] Karaklajic, D., Fan, J., Schmidt, J-M., Verbauwhede, I.: Low-cost fault detection method for ECC using Montgomery powering ladder. DATE 2011, IEEE, 1, pp. 1–6, (2011).

[77] Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation, vol. 48, 177, pp. 203–209, (1987).

[78] Kocher, P., Jaffe, J., Jun, B.: Differential power analysis, CRYPTO 1999, LNCS 1666, Springer, pp. 388–397, (1999).

[79] Kim, C-K., Ha, J-C., Moon, S-J., Yen, S-M., Lien, W-Ch., Kim, S-H.: An improved and efficient countermeasure against power analysis attacks. *https* : *//eprint.iacr.org*/2005/022.*pdf*, IACR Cryptology ePrint Archive, (2005).

[80] Kocher. P: Timing attacks on implementation of Diffie-Hellman RSA, DSS and other systems. CRYPTO 1996, LNCS 1109, Springer, pp. 104–113, (1996).

[81] Koç, Ç. K.: Cryptographic engineering. Springer-Verlag, (2009).

[82] Longa, P., Miri, A.: Fast and flexible elliptic curve point arithmetic over prime fields. IEEE Transactions on Computers, vol. 57, (3), pp. 289–302, (2008).

[83] Langley, A., Hamburg, M., Turner, S.: Elliptic curves for security, IRTF. https://tools.ietf.org/html/rfc7748, (2016).

[84] Liardet, P.-Y., Smart, N.P.: Preventing SPA/DPA in ECC systems using the Jacobi form. CHES 2001, LNCS 2162, Springer, pp. 391–401, (2001).

[85] Le, D-P., Tan, C-H., Tunstall, M.: Randomizing the Montgomery powering ladder. IFIP 2015, LNCS 9311, Springer, pp. 169–184, (2015).

[86] Liu, Z., Liu, D., Sun, X., Zou, X., Lin, H.: Implementation of a resource-constrained ECC processor with power analysis countermeasure. APCCAS 2016, Springer, pp. 206–209, (2017).

[87] López, J., Dahab, R.: Fast multiplication on elliptic curves over GF(2m) without precomputation. CHES 1999, LNCS 1717, Springer, pp. 316–327, (1999).

[88] Mangard, S., Oswald, E., Popp, T.: Power analysis attacks: Revealing the secrets of smart cards, Springer-Verlag. (2007).

[89] Miller, V.: Use of elliptic curves in cryptography. CRYPTO 1985, LNCS 218, Springer, pp. 417–426, (1985).

[90] Miret, J., Sadornil, D., Tena, J., Tomàs, R., Valls, M.: On avoiding ZVP-attacks using isogeny volcanoes. WISA 2008, LNCS 5379, Springer, pp. 266–277, (2009).

[91] Muller, F., Valette, F.: High-order attacks against the exponent splitting protection. PKC 2006, LNCS 3958, Springer, pp. 315–329, (2006).

[92] Mamiya, H., Miyaji, A., Morimoto, H.: Secure elliptic curve exponentiation against RPA, ZRA, DPA, and SPA, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E89-A, 8, pp. 2207–2215, (2006).

[93] Mamiya, H., Miyaji, A., Morimoto, H.: Efficient countermeasures against RPA, DPA, and SPA. CHES 2004, LNCS 3156, Springer, pp. 343–356, (2004).

[94] Montgomery, P. L.: Speeding the Pollard and elliptic curve methods of factorization. Mathematics of Computation, vol. 48, 177, pp. 243–264, (1987).

[95] Möller, B.: Securing elliptic curve point multiplication against side-channel attacks. ISC 2001, LNCS 2200, Springer, pp. 324–334, (2001).

[96] Murdica, C., Guilley, S., Danger, J-L., Hoogvorst, Ph., Naccache, D.: Same values power analysis using special points on elliptic curves. COSADE 2012, LNCS 7275, Springer, pp. 183–198, (2012).

[97] Murdica, C.: Physical security of elliptic curve cryptography. Telecom ParisTech, https://pastel.archives-ouvertes.fr/tel-01179584/document (2014).

[98] Nascimento, E., Abarzúa, R., López, J., Dahab, R.: A comparison of simple side-channel analysis countermeasures for variable-base elliptic curve scalar multiplication. XIV Simposio Brasilerio em Seguranca da Informacao de Sistemas Computacionais, SBseg 2014, pp. 125–138, (2014).

[99] Naccache, D., Smart, N., Stern, J.: Projective coordinates leak. EUROCRYPT 2004, LNCS 3027, Springer, pp. 257–267, (2004).

[100] Nascimento, E., Chmielewski, Ł., Oswald, D., Schwabe, P.: Attacking embedded ECC implementations through cmov side channels. SAC 2016, LNCS 10532, Springer, pp. 99–119, (2017).

[101] NIST.: FIPS 186-3: Digital signature standard (DSS), National institute of standards and yechnology NIST, https://csrc.nist.gov/publications/detail/fips/186/3/archive/2009-06-25, (2009). Accessed 28 June 2018.

[102] Okeya, K., Sakurai, K.: Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. INDOCRYPT 2000, LNCS 1977, Springer, pp. 178–190, (2000).

[103] Quisquater, J-J., Samyde, D.: Electromagnetic analysis (EMA): Measures and coutermeasures for smard cards. Smart Card Programming and Security – E-SMART 2001, LNCS 2140, Springer, pp. 200–210, (2001).

[104] Popp, Th., Mangard, S., Oswald, E.: Power analysis attacks and countermeasures. IEEE Design & Test of Computers, vol. 24, 6, pp. 535–543, (2007).

[105] Prouff, E., Rivain, M.: Theoretical and practical aspects of mutual information based side channel analysis. ACNS 2009, LNCS 5536, Springer, pp. 499–518, (2009).

[106] Perin, G., Imbert, L., Torres, L., Maurine, Ph.: Attacking randomized exponentiations using unsupervised learning. COSADE 2014, LNCS 8622, Springer, pp. 144–160, (2014).

[107] Joye, M., Quisquater, J-J.: Hessian elliptic curves and side-channel attacks. CHES 2001, LNCS 2162, Springer, pp. 402–410, (2001).

[108] Renes, J., Costello, C., Batina, L.: Complete addition formulas for prime order elliptic curves. EUROCRYPT 2016, LNCS 9665, Springer, pp. 403–428, (2016).

[109] Clavier, Ch., Feix, B., Gagnerot, G., Giraud, Ch., Roussellet, M., Verneuil, V.: ROSETTA for single trace analysis. INDOCRYPT 2012, LNCS 7668, Springer-Verlag, pp. 140–155, (2012).

[110] Research center for information security: Side-channel attack standard evaluation board (SASEBO), http://satoh.cs.uec.ac.jp/SASEBO/en/board/index.html", accessed January 2018. (2016).

[111] Schramm, K. and Wollinger, Th. and Paar, Ch.: A new class of collision attacks and its application to DES. FSE 2003, LNCS 2887, Springer, pp. 206–222, (2003).

[112] Smart, N.: An analysis of Goubin's refined power analysis attack. CHES 2003, LNCS 2779, Springer, pp. 281–290, (2003).

[113] Smart, N.: The Hessian form of an elliptic curve. CHES 2001, LNCS 2162, Springer, pp. 118–125, (2001).

[114] Brown, D. R. L.: STANDARDS FOR EFFICIENT CRYPTOGRAPHY, SEC 2: Recommended elliptic curve domain parameters. http://www.secg.org/sec2-v2.pdf, Certicom Corp. Version 2.0, January (2010),

[115] Stebila, D., Thériault, N.: Unified point addition formulæ and side-channel attacks. CHES 2006, LNCS 4249, Springer, pp. 354–368, (2006).

[116] Stankovic, J. A.: Research directions for the internet of things. IEEE Internet of Things Journal, Vol. 1, (1), pp. 3–9, (2014).

[117] Strobel, D., Oswald, D., Richter, B., Schellenberg, F., Paar, Ch.: Microcontrollers as in security devices for pervasive computing applications. Proceedings of the IEEE, vol. 102, 8, pp. 1157–1173, (2014).

[118] Schmidt, J-M., Tunstall, M., Avanzi, R., Kizhvatov, I., Kasper, T., Oswald, D.: Combined implementation attack resistant exponentiation. LATINCRYPT 2010, LNCS 6212, Springer, pp. 305–322, (2010).

[119] Specht, R., Heyszl, J., Kleinsteuber, M., Sigl, G.: Improving non-profiled attacks on exponentiations based on clustering and extracting leakage from multi-channel high-resolution EM measurements. COSADE 2015, LNCS 9064, Springer, pp. 3–19, (2015)

[120] Standaert, F-X., Gierlichs, B., Verbauwhede, I.: Partition vs. comparison side-channel distinguishers: An empirical evaluation of statistical tests for univariate side-channel attacks against two unprotected CMOS devices. ICISC 2008, LNCS 5461, Springer, pp. 253–267, (2009).

[121] EStrauss, E. G.: Addition chains of vectors (problem 5125). American Mathematical Monthly, Vol. 70, pp. 806–808, (1964).

[122] Smart, N., Oswald, E., Page, D.: Randomised representations. IET Information Security, vol. 2, 2, pp. 19–27, (2008).

[123] Sato, H., Schepers, D., Takagi, T.: Exact analysis of Montgomery multiplication. INDOCRYPT 2004, LNCS 3348, Springer, pp. 290-304, (2004).

[124] Tunstall, M., Joye, M.: Coordinate blinding over large prime fields. CHES 2010, LNCS 6225, Springer, pp. 443–455, (2010).

[125] Trichina, E., Belleza, A.: Implementation of elliptic curve cryptography with built-in counter measures against side channel attacks. CHES –2002, LNCS 2523, Springer, pp. 98–113, (2002).

[126] Thériault, N.: SPA resistant left-to-right integer recoding. SAC 2005, LNCS 3897, Springer, pp. 345–358, (2005).

[127] Walter, C. D.: Sliding windows succumbs to big mac attack. CHES 2001, LNCS 2162, Springer, pp. 286–299, (2001).

[128] Yen, S-M., Ko, L-C., Moon, S-J., Ha, J-C.: Relative doubling attack against Montgomery ladder: ICISC 2005, LNCS 3935, Springer, pp. 117–128, (2006).

[129] Yen, S-M., Lien, W-C., Moon, S-J., Ha, J-C.: Power analysis by exploiting chosen message and internal collisions vulnerability of checking mechanism for RSA-decryption. Mycrypt 2005, LNCS 3715, Springer, pp. 183–195, 2005.

[130] Yen, S-M., Joye, M.: Checking before output may not be enough against fault-based cryptanalysis. IEEE Transactions on Computers, Vol. 49, 9, pp. 967–970, (2000).

[131] Yen, S-M., Kim, S., Lim, S., Moon, S-J.: A Countermeasure against one physical cryptanalysis may benefit another attack. ICISC 2001, Springer, LNCS 2288, pp. 269–294, (2002).

[132] Venelli, A., Contribution à la sécurité physique des cryptosystèmes embarqués. Université Aix-Marseille, Thèse Docteur de Informatique, (2011).

[133] Verneuil, V.: Cryptographie à base de courbes elliptiques et sécurité de composants embarqués. Université de Bordeaux, Thèse École Doctorale de Mathématiques et Informatique, (2012).

[134] Walter, C. D.: Simple power analysis of unified code for ECC double and add. CHES-2004, LNCS 3156, Springer-Verlag, pp. 191–204, (2004).

[135] Wang, L., Li, Q., Zhang, G., Yu, J., Zhang, Z., Guo, L., and Zhang, D.: A new SPA attack on ECC with regular point multiplication. CIS 2015, IEEE, pp. 322–325, (2016).

[136] Washington, L.C.: Elliptic curves number theory and cryptography. Chapman and Hall/CRC, Discrete Mathematics Series, (2008).

[137] Wenger, E., Großschädl, J.: An 8-bit AVR-based elliptic curve cryptographic RISC processor for the internet of things. MICROW 2012, IEEE, pp. 39–46, (2012).