Dynamic Countermeasure Against the Zero Power Analysis

Jean-Luc Danger[1,2], Sylvain Guilley[1,2], Philippe Hoogvorst[2], Cédric Murdica[1,2], and David Naccache[3]

[1] Secure-IC S.A.S., 80 avenue des Buttes de Coësmes, F-35700 Rennes, France
{jean-luc.danger, sylvain.guilley, cedric.murdica}@secure-ic.com
[2] Département COMELEC, Institut TELECOM, TELECOM ParisTech, CNRS LTCI, Paris, France
{jean-luc.danger, sylvain.guilley, philippe.hoogvorst, cedric.murdica}@telecom-paristech.fr
[3] École normale supérieure, Département d'informatique 45, rue d'Ulm, F-75230, Paris Cedex 05, France.
david.naccache@ens.fr

Abstract. Elliptic Curve Cryptography can be vulnerable to Side-Channel Attacks, such as the Zero Power Analysis (ZPA). This attack takes advantage of the occurrence of special points that bring a zero-value when computing a doubling or an addition of points. This paper consists in analysing this attack. Some properties of the said special points are explicited. A novel dynamic countermeasure is described. The elliptic curve formulæ are updated depending on the elliptic curve and the provided base point.

Keywords: Elliptic Curve Cryptography, Side-Channel Analysis, Zero Power Analysis, Zero-Value Points, Dynamic Countermeasure, Jacobi Symbol.

1 Introduction

Elliptic Curve Cryptography (ECC) is vulnerable to the Correlation Power Analysis [5, §3.2]. Randomizing the base point, such as the Random Projective Coordinates [5, §5.3] and the Random Curve Isomorphism [12], is an efficient way to prevent the CPA.

However, these countermeasures are not enough because of some refined attacks such as the Refined Power Analysis (RPA), introduced by Goubin [9], and its extension, the Zero Power Analysis (ZPA), introduced by Akishita and Takagi [1]. The RPA takes advantage of the occurrence or the absence of particular points of the form $(0, y)$. These points are randomized by neither the Random Projective Coordinates nor the Random Curve Isomorphism. The ZPA does not focus only on a zero-value in points' coordinates, but also on a possible zero-value in intermediate variables when computing a doubling or an addition of points. Such particular points are defined as *zero-value points* [1]. The RPA becomes a particular case of the ZPA.

This paper is an analysis of these attacks. Some properties of the zero-value points are given. These properties are valuable, they allow performing some verifications at the beginning of the Elliptic Curve Scalar Multiplication. The elliptic curve formulæ are adapted according to the given elliptic curve and the given base point for a protection against the ZPA.

The rest of the paper is structured as follows. Section 2 briefly recalls on ECC and on the RPA and ZPA attacks. Section 3 is devoted to the properties of the zero-value points. Section 4 gives some existing methods to prevent the RPA and the ZPA. These methods consist in modifying the formulæ so that a zero-value point never occurs. This decreases the performance since more field operations are required for performing doubling or addition of points. In Section 5, we expose new methods to prevent the ZPA, including:

- the dynamical check that the given curve does not contain any zero-value point for doubling; the appropriate formulæ are chosen in consequence,
- the modification of the base point, so that the absence of zero-value points for addition is ensured during the computation of the ECSM.

Finally, we conclude in Section 6.

2 Preliminaries

This section gives the notions on ECC and describe the attacks RPA and ZPA. This is required to fully understand the next sections.

2.1 Elliptic Curve Cryptography

An elliptic curve over a finite prime field $\mathbb{F}_p$ of characteristic $p > 3$ can be described by its reduced Weierstraß form:

$$E\colon y^2 = x^3 + ax + b \ . \tag{1}$$

with $a, b \in \mathbb{F}_p$ verifying $4a^3 + 27b^2 \neq 0$. We denote by $E(\mathbb{F}_p)$ the set of points $(x, y) \in \mathbb{F}_p^2$ satisfying equation (1), plus the point at infinity $\mathcal{O}$.

$E(\mathbb{F}_p)$ is an additive abelian group defined by the following addition law. Let $P_1 = (x_1, y_1) \neq \mathcal{O}$ and $P_2 = (x_2, y_2) \notin \{\mathcal{O}, -P_1\}$ be two points on $E(\mathbb{F}_p)$. Point addition $P_3 = (x_3, y_3) = P_1 + P_2$ is defined by the formula:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \quad \text{where } \lambda = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} & \text{if } P \neq Q, \\ \frac{3x_1^2 + a}{2y_1} & \text{if } P = Q. \end{cases}$$

The inverse of point P_1 is defined as $-P_1 = (x_1, -y_1)$.

ECC relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP, compute k given P and $Q = [k]P$) or on the hardness of related problems such as ECDH or ECDDH, which can be solved if ECDLP can be.

2.2 Jacobian Projective Arithmetic

The equation of an elliptic curve in the Jacobian projective coordinates system in the reduced Weierstraß form is:

$$E^{\mathcal{J}}\colon Y^2 = X^3 + aXZ^4 + bZ^6 \ .$$

The projective point (X, Y, Z) corresponds to the affine point $(X/Z^2, Y/Z^3)$ and there is an equivalence relation between the points: the point (X, Y, Z) is equivalent to any point

(r^2X, r^3Y, rZ) for all $r \in \mathbb{F}_p^*$. The point at infinity is defined as $\mathcal{O} = (1, 1, 0)$ in Jacobian coordinates.

We give addition (ECADD) and doubling (ECDBL) formulas in the Jacobian projective coordinates system. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ two points of $E^{\mathcal{J}}(\mathbb{K})$.

- **ECDBL**. $P_3 = (X_3, Y_3, Z_3) = 2P_1$ is computed as:
  $X_3 = T,\ Y_3 = -8Y_1^4 + M(S - T),\ Z_3 = 2Y_1Z_1$, where
  $S = 4X_1Y_1^2,\ M = 3X_1^2 + aZ_1^4,\ T = -2S + M^2$;
- **ECADD**. $P_3 = (X_3, Y_3, Z_3) = P_1 + P_2$ is computed as:
  $X_3 = -H^3 - 2U_1H^2 + R^2,\ Y_3 = -S_1H^3 + R(U_1H^2 - X_3),\ Z_3 = Z_1Z_2H$, where
  $U_1 = X_1Z_2^2,\ U_2 = X_2Z_1^2,\ S_1 = Y_1Z_2^3,\ S_2 = Y_2Z_1^3,\ H = U_2 - U_1,\ R = S_2 - S_1$.

ECDBL needs 4 multiplications, 6 squares and 7 additions/subtractions. ECADD needs 12 multiplications, 4 squares and 7 additions/subtractions.

Many different formulæ exist in the literature, such as the mixed coordinates [4] or the co-Z formulæ [13,10].

2.3 Elliptic Curve Scalar Multiplication

In ECC applications, one has to compute scalar multiplications (ECSMs), *i.e.* compute $[k]P$, given P and an integer k. The Double-and-Add always method (Algorithm 1), secure against the Simple Power Analysis [5], can be used to perform such a computation.

Algorithm 1 Double-and-Add always [5, §3.1]

Input: $k = (1, k_{n-2}, \ldots, k_0)_2$, P
Output: $[k]P$
  $R[0] \leftarrow P$
  $R[1] \leftarrow P$
  for $i = n - 2$ **downto** 0 **do**
    $R[0] \leftarrow 2R[0]$
    $R[1 - k_i] \leftarrow R[0] + P$
  end for
  return $R[0]$

Applying the Double-and-Add always using ECDBL and ECADD requires $16n$ multiplications, $10n$ squares and $14n$ additions/subtractions.

2.4 Refined Power Analysis

The Refined Power Analysis (RPA) introduced by Goubin [9] is based on the occurrence of the particular point $P_0 = (0, y)$ during the ECSM. The attacker chooses the base point P such that the special point P_0 will occur on certain assumptions (for example the current targeted bit of key k is 0). The computation of such a point P is performed as follows, with the example of the Double-and-Add always method (Algorithm 1).

The attack is recursive. Suppose that the attacker already knows the $n - i - 1$ leftmost bits of the fixed scalar $k = (k_{n-1}, \ldots, k_0)_2$ and tries to recover k_i. The attacker computes

the point $P = [(k_{n-1}, \ldots, k_{i+1}, 1)_2^{-1} \bmod \#E]P_0$. The point P_0 will be doubled at iteration $i-1$ only if $k_i = 1$.

The doubling of the point P_0 can easily be detected by observing the trace, as shown in Figure 1.

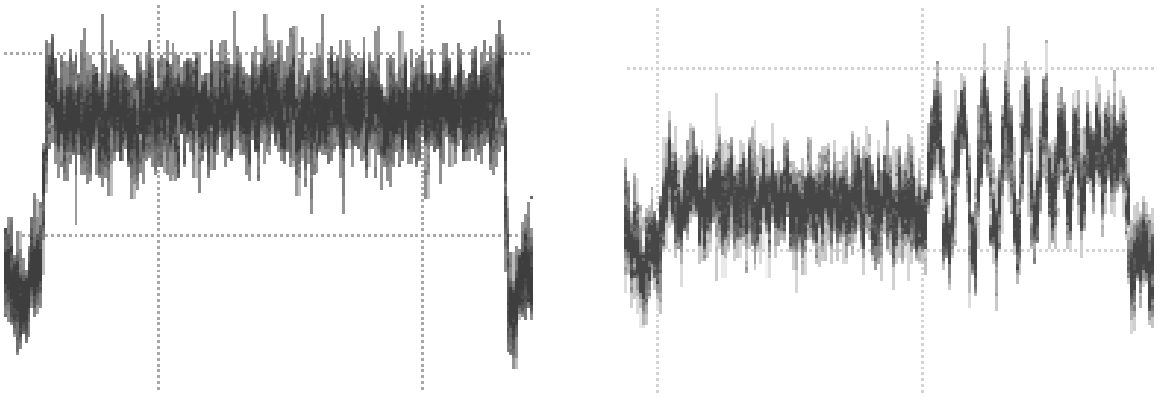

Fig. 1. Power consumption of modular multiplications of two random operands (left curve) and a random operand and zero (right curve)

2.5 Zero Power Analysis

The Zero Power Analysis (ZPA) of Akishita and Takagi [1] is an extension of the RPA. This attack does not only focus on points with a zero x coordinate but in intermediate values that can possibly take the value zero when performing a doubling or an addition. Such points are called *zero-value points*. An elliptic curve does not necessarily contain a point of the form $(0, y)$. In this case, the RPA cannot be applied. The ZPA brings more possible special points, and therefore can be applied to a larger set of elliptic curves.

The zero-value points depend on the formulæ used (see Section 2.2). They also depend on the way it is computed. For example, for the formula ECADD, the value X_3 can be computed in different manners by changing the order of additions and subtractions. The different ways are analysed in [1]. In fact, we will see in Section 4 that the way X_3 is computed does not matter. A simple method permits to avoid zero-values whatever the order of additions and subtractions without any extra multiplication. We only list the conditions where a zero-value is an input of a multiplication.

On the doubling formula given in Section 2.2, the intermediate values that can take one zero-value are $X_1, X_3 = T = -S + M^2, M, S$. In affine coordinates, this corresponds to the following conditions:

- $x_1 = 0$ (D1), this corresponds to $X_1 = 0$ and thus $S = 0$,
- $x_3 = 0$ (D2), this corresponds to $X_3 = T = 0$,
- $3x_1^2 + a = 0$ (D3), this corresponds to $M = 0$.

Remark 1. $Y_1, Y_3, Z_1, S - T$ cannot be equal to zero because this would mean that the point doubling or the result point is the point at infinity ($Z_1 = 0$) or has low order ($Y_1 = 0 \Rightarrow P_1$

has order 2; $Y_3 = 0 \Rightarrow P_3$ has order 2; $S - T = 0 \Rightarrow x_1 = x_3 \Rightarrow 2P_1 = \pm P_1 \Rightarrow P_1$ has order 3 or $P_1 = \mathcal{O}$), which is impossible when computing the ECSM of the base point [1].

On the addition formula given in Section 2.2, the intermediate values that can take the value zero are X_1, X_2, X_3, R. In affine coordinates, this corresponds to the following conditions:

- $x_1 = 0$ (A1), this corresponds to $X_1 = 0$,
- $x_2 = 0$ (A2), this corresponds to $X_2 = 0$,
- $x_3 = 0$ (A3), this corresponds to $X_3 = 0$,
- $y_2 - y_1 = 0$ (A4), this corresponds to $R = 0$.

Remark 2. $Y_1, Z_1, Y_2, Y_3, Z_2, H, U_1H^2 - X_3$ cannot be equal to zero because this would mean that one of the points is the point at infinity ($Z_1 = 0$; $Z_2 = 0$) or one of the points has order 2 ($Y_1 = 0$; $Y_2 = 0$; $Y_3 = 0$) or $P_1 = \pm P_2$ ($H = 0 \Rightarrow x_1 = x_2 \Rightarrow P_1 = \pm P_2$) or $P_3 = \pm P_1$ ($U_1H^2 - X_3 = 0 \Rightarrow x_1 = x_3$), which is impossible when computing the ECSM of the base point [1].

Remark 3. The conditions where the x coordinate is zero corresponds to the RPA attack.

For both addition and doubling, the mixed coordinates [4] and the co-Z formulæ [13,10] bring the same conditions. This is because the numerator of λ of the formulæ in affine coordinates is always computed.

2.6 Finding zero-value points

We give some methods for finding zero-value points to perform the ZPA, given an elliptic curve. Let us take for example the Double-and-Add always algorithm (Algorithm 1). Of course, the method can be adapted to other ECSMs. Suppose that the attacker already knows the $n - i - 1$ leftmost bits of the fixed scalar $k = (k_{n-1}, \ldots, k_0)_2$ and tries to recover k_i. The attacker has several possibilities, listed hereafter.

Taking advantage of condition (D1). If the given elliptic curve contains a point of the form $P_0 = (0, y)$, the attacker can compute the point $P = [(k_{n-1}, \ldots, k_{i+1}, 1)_2^{-1} \bmod \#E]P_0$. The point P_0 will be doubled only if $k_i = 1$. Taking advantage of conditions (D2), (A1), (A2) or (A3) is similar.

Taking advantage of condition (D3). If the given elliptic curve contains a point $P_1 = (x, y)$ such that $3x^2 + a = 0$ the attacker can compute the point $P = [(k_{n-1}, \ldots, k_{i+1}, 1)_2^{-1} \bmod \#E]P_1$. The point P_1 will be doubled only if $k_i = 1$.

Taking advantage of condition (A4). It is a bit more tricky. Indeed, the attacker has to find a base point $P = (x_P, y_P)$ such that $Q = (x_Q, y_Q) = [(k_{n-1}, \ldots, k_{i+1}, 1)_2]P$ and P satisfy $y_Q - y_P = 0$. The best known method to find such a point P is to use the division polynomials [16, §3.2]. For a positive integer m, if $[m]P \neq \mathcal{O}$, $[m]P$ can be expressed as

$$[m]P = \left(\frac{\phi_m(x)}{\psi_m(x)^2}, \frac{\omega_m(x, y)}{\psi_m(x, y)^3} \right)$$

where ψ_m denotes the m^{th} division polynomial, and is recursively computed[4]. ϕ_m and ω_m are defined as

$$\phi_m = x\psi_m^2 - \psi_{m+1}\psi_{m-1} \ ,$$
$$\omega_m = \frac{\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2}{4y} \ .$$

Denote $m = (k_{n-1}, \ldots, k_{i+1}, 1)_2$. Finding a point $P = (x_P, y_P)$ such that $Q = (x_Q, y_Q) = [m]P$ and P satisfy $y_Q - y_P = 0$, can be done as follows. First, find x_P a solution of the following equation.

$$f(x, y) = \omega_m(x, y) - y\psi_m(x, y)^3 = 0 \ . \tag{2}$$

Using the recursive properties of the division polynomials, and by replacing y^2 by x^3+ax+b, $f \in \mathbb{Z}[x]$ if m is even, and $\frac{f}{y} \in \mathbb{Z}[x]$ if m is odd [16, §3.2]. If $x_P^3 + ax_P + b$ is a square in $\mathbb{F}_p$, then $y_P = \sqrt{x_P^3 + ax_P + b}$.

Solving Equation 2 for large m is hard. Indeed, it consists in finding roots of polynomials of degree $m^2 + m$ which is difficult [1].

However, it is still feasible for small m. In addition, there is no guarantee that there is no other more efficient method to compute such a point. The protection against ZPA presented in this paper is ensured.

2.7 Isogeny defence

To protect against the RPA, Smart introduced the isogeny defence [14]. It consists in computing the ECSM on an isogenous curve E' that does not contain any point of the form $(0, y)$, instead of the given elliptic curve E. Akishita and Takagi extended later the countermeasure to also prevent the ZPA [2].

Isogenous curves of standardized curves are precomputed and stored in the chip. Indeed, finding isogenies is not trivial and cannot be done on the fly with a new given elliptic curve. The countermeasure proposed in this paper is dynamic and works on any curve.

2.8 Scalar Randomization

Randomizing the scalar, such as the group scalar randomization [5, §5.1], the additive splitting [6, §4.2], the Euclidean splitting [3, §4] or the multiplicative splitting [15] are believed to be secure against the RPA and ZPA.

However, as opposed to the isogeny defence and our proposed methods, the absence of special points is not ensured with scalar randomization techniques. Since several bits can be targeted at a time with the RPA and the ZPA, the attacker can recover several bits of the randomized scalar. The system is not fully broken, nevertheless the security is compromised.

3 Properties of the Zero-Value Points

In this section, we give some properties of the zero-value points, namely, points satisfying (D1), (D3) and (A4). Given the elliptic curve $E\colon y^2 = x^3 + ax + b$, defined over $\mathbb{F}_p$, we will see how to verify whether the curve does contain zero-value points or how to tell if a given point satisfies condition (A4).

[4] The recursive process to compute the division polynomials is given in Appendix

3.1 $P = (0, y_P)$ (D1)

We define the particular points with a zero x coordinate.

Definition 1. *A point $P \in E$ of the form $P = (0, y_P)$ is called a zero x-coordinate point.*

Proposition 1. *E contains zero x-coordinate points over $\mathbb{F}_p$ iff b is a square. In this case, the zero x-coordinate points are $(0, \sqrt{b})$ and $(0, -\sqrt{b})$.*

Proof. ($\Longrightarrow$) Suppose E contains a zero x-coordinate point $P = (0, y_P)$, then $(0, y_P)$ satisfies the curve equation and $y_P^2 = b \implies b$ is a square. ($\Longleftarrow$) If $b = y_P^2$ for some $y_P \in \mathbb{F}_p$, then the pair $(0, y_P) \in \mathbb{F}_p^2$ satisfies the curve equation and therefore lies on the curve E. In this case, $(0, -y_P)$ also lies on the curve. □

3.2 $P = (x_P, y_P)$ satisfying $3x_P^2 + a = 0$ (D3)

Definition 2. *A point $P = (x_P, y_P) \in E$ satisfying $3x_P^2 + a = 0$ is called a zero tangent line slope point.*

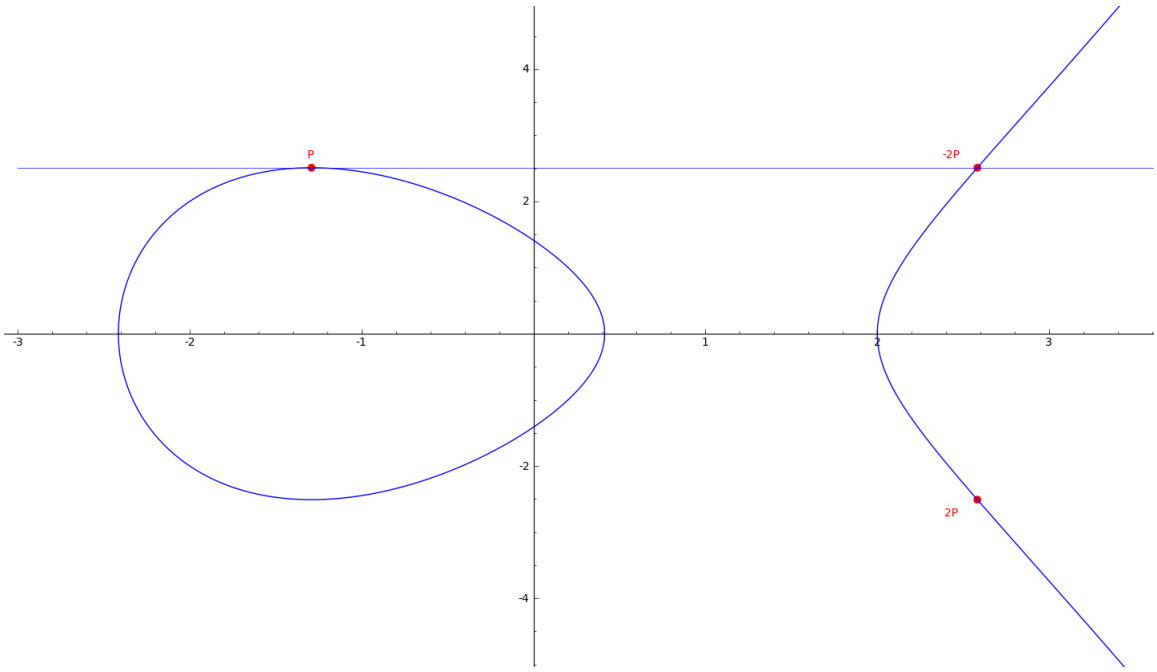

Fig. 2. A *zero tangent line slope point* on the curve $y^2 = x^3 - 5x + 1$ over $\mathbb{R}$

Proposition 2. *E contains zero tangent line slope points over $\mathbb{F}_p$ iff the two following conditions are satisfied*

1. *$-3a$ is a square, denote $\delta = \sqrt{-a/3}$,*
2. *$-\frac{a}{3}\delta + a\delta + b$ or $\frac{a}{3}\delta - a\delta + b$ is a square.*

Proof. ($\Longrightarrow$) Suppose E contains a zero tangent line slope point $P = (x_P, y_P)$. Then, x_P verifies $3x_P^2 + a = 0 \implies -a/3$ is a square, and therefore $-9\frac{a}{3} = -3a$ is a square because 9 is. x_P can take two values:

- $x_P = \sqrt{-a/3} = \delta$, in this case, $y_P^2 = x_P^3 + ax_P + b = -\frac{a}{3}\delta + a\delta + b$ is a square, or
- $x_P = -\sqrt{-a/3} = -\delta$, in this case, $y_P^2 = x_P^3 + ax_P + b = \frac{a}{3}\delta - a\delta + b$ is a square.

($\Longleftarrow$) Suppose $-3a$ is a square and denote $\delta = \sqrt{-a/3}$. If $-\frac{a}{3}\delta + a\delta + b$ is a square, then the pair (x_P, y_P), with $x_P = \delta, y_P = \sqrt{-\frac{a}{3}\delta + a\delta + b}$, satisfies the curve equation and therefore (x_P, y_P) lies on the curve. Moreover x_P satisfies $3x_P^2 + a = 0$. If $\frac{a}{3}\delta - a\delta + b$ is a square, then the pair (x_P, y_P), with $x_P = -\delta, y_P = \sqrt{\frac{a}{3}\delta - a\delta + b}$ satisfies the curve equation and therefore (x_P, y_P) lies on the curve. Moreover x_P satisfies $3x_P^2 + a = 0$. □

From the proposition, we can give the following trivial corollary.

Corollary 1. *If $-3a$ is not a square, E does not contain any zero tangent line slope point.*

3.3 $P = (x_P, y_P), Q = (x_Q, y_Q)$ satisfying $y_P - y_Q = 0$ (A4)

Definition 3. *A point $P = (x_P, y_P) \in E$ such that there exists a point $Q = (x_Q, y_Q) \in E$, with $Q \neq P$ and $y_P = y_Q$ is called a y same coordinate point.*

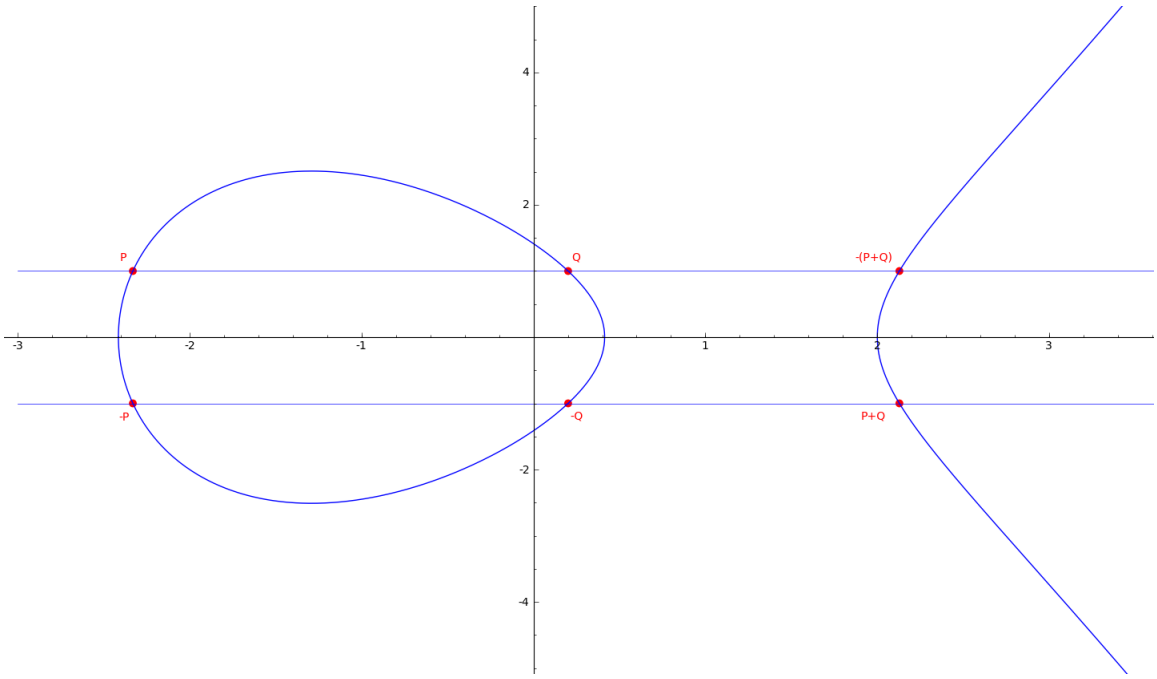

Fig. 3. Some y same coordinate points on the curve $y^2 = x^3 - 5x + 1$ over $\mathbb{R}$

Proposition 3. *Let $P = (x_P, y_P)$ a point of E with order different from 3. P is a y same coordinate point iff $-3x_P^2 - 4a$ is a square.*

Proof. Let $P = (x_P, y_P) \in E$. By Bézout's theorem, the line $y = y_P$ has one or three intersections, counting with multiplicity, with the curve E. Finding the intersections can be done by solving the equation $y_P^2 = x^3 + ax + b$. x_P is a solution, thus the equation is equivalent to (by dividing by $x - x_P$):

$$x^2 + x_P x + x_P^2 + a = 0 \ . \tag{3}$$

This equation has two roots, counting with multiplicity, iff the discriminant $-3x_P^2 - 4a$ is a square. Moreover, at least one of the root is different to x_P, otherwise this would mean that P is the intersection of the line $y = y_P$ and E with multiplicity 3. By the addition law, this would mean that $P + P = -P \Rightarrow [3]P = \mathcal{O}$ which is impossible by the hypothesis of P. □

With this proposition, the following corollary defines a set of elliptic curves that do not contain any y same coordinate point.

Corollary 2. *If $a = 0$ and -3 is not a square in $\mathbb{F}_p$, then E does not contain any y same values point of order different from 3.*

Proof. Suppose that $a = 0$ and -3 is not a square in $\mathbb{F}_p$, and $P = (x_P, y_P) \in E$ is a y same coordinate point, thus $-3x_P^2$ is a square and therefore -3 is square. By contradiction, E does not contain any y same coordinate point. □

Remark 4. On the other hand, if $a = 0$ and -3 is a square, all points are y same coordinate points.

4 Modifying Formulæ

Some countermeasures consist in modifying the formulæ so that a zero value is never manipulated. This prevent against the RPA and ZPA.

Itoh et al. introduced the Random Linear Coordinates [11]. It consists in replacing the point $P = (X, Y, Z)$ by (X', Y, Z, μ) with μ a random element in $\mathbb{F}_p$ and $X' = X + \mu$, to avoid direct manipulation of X. The modified formulæ are protected against zero x-coordinate and zero tangent line slope points (they omit the y same coordinate points).

Danger et al. proposed an alternative solution to prevent against zero x-coordinate points [7]. It consists in modifying the given elliptic curve using an isomorphism to transform the base point $P = (x, y)$ into $P' = (0, y')$. The given elliptic curve E is mapped to a curve E' of the form $y^2 = x^3 + a_2x^2 + a_4x + a_6$. E' is not in the Weierstraßform. Therefore, an adaptation of the formulæ has to be done. The impact of the countermeasure is reduced because the addition with P' is simplified due to the zero value. When using co-Z formulæ, this does not bring any additional cost. We adapted the method to the formulæ of Section 2.2 (ECDBL and ECADD), mixed with the methods described below. It turns out it is more efficient than the Random Linear Coordinates for protecting against zero x-coordinate and zero tangent line slope points [5], only on the case where the base point, or its opposite ($-P' = (0, -y)$) is frequently manipulated during the ECSM, like the Double-and-Add always method (Algorithm 1) and the Montgomery Ladder using co-Z formulæ [10, Algorithm 7].

[5] We first omit the y same coordinate points, for a comparison to the Random Linear Coordinates

We give some very trivial methods to modify the formulæ so that a zero-value can never occur. In the following, r denotes a random element of $\mathbb{F}_p$.

Protecting an additions' sequence (method 1). Denote $v \leftarrow \sum_{i=0}^{m}(-1)^{\mu_i}u_i$ a sequence of additions and subtractions, with $m > 0, \mu_i \in \{0,1\}, u_i \in \mathbb{F}_p$. In order to prevent from possible zero-values that can occur during the sequence, one can simply perform the following additions' sequence $t \leftarrow r + \sum_{i=0}^{m}(-1)^{\mu_i}u_i$ instead, and start the sequence with the addition of r. Then, compute $u = t - r$ to recover the correct value u. The cost of the protection is 2 additions[6].

Protecting a square (method 2). A method to protect a square u^2, with u possibly equals zero, is to compute $(u+r)(u-r) = u^2 - r^2$. The correct value u^2 can be recovered by subtracting r^2. If r^2 is precomputed, the cost of the protection is 3 additions[6].

Protecting a multiplication (method 3). A method to protect a multiplication uv, with u possibly equals zero, and $v \neq 0$, is to compute $s = (u+r)v = uv + rv$ and $t = rv$. The true value uv can be recover later by computing $s - t$. In this case, the cost of the protection is 1 multiplication and 2 additions[6].

We give addition (ECADD-D1) and doubling (ECDBL-D1-D3)[7] formulas in the Jacobian projective coordinates system. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ two points of $E\colon y^2 = x^3 + a_2x^2 + a_4x + a_6$. We recall that, when using the method described in [7], the addition of points is always performed with a zero x coordinate.

Algorithm 2 ECDBL-D1-D3

Input: $P = (X_1, Y_1, Z_1) \in E^{\mathcal{J}}, r, s = r^2$
Output: $2P$
$S \leftarrow 4X_1Y_1^2;\ M' \leftarrow r + 3X_1^2 + 2a_2X_1Z_1^2 + a_4Z_1^4$
$M'' \leftarrow M' - 2r;\ Z_3 \leftarrow 2Y_1Z_1$
$T' \leftarrow r + M'M'' - 2S - a_2Z_3^2 - s$
$X_3 \leftarrow T' - r$
$Y_3' \leftarrow r - 8Y_1^4 + M'(S - X_3) - r(S - X_3)$
$Y_3 \leftarrow Y_3 - r$
return (X_3, Y_3, Z_3)

[6] By convention, and for a better clarity, we set 1 addition = 1 subtraction in terms of computational cost
[7] D1-D3 means that it is protected against zero x-coordinate (condition D1) and zero tangent line slope points (condition D3)

Algorithm 3 ECADD-D1

Input: $P = (0, Y_1, Z_1), Q = (X_2, Y_2, Z_2) \in E^{\mathcal{J}}$,
$r, s = r^2$
Output: $P + Q$
$H \leftarrow X_2 Z_1^2$
$S_1 \leftarrow Y_1 Z_2^3$; $S_2 \leftarrow Y_2 Z_1^3$
$R \leftarrow S_2 - S_1$; $Z_3 \leftarrow Z_1 Z_2 H$
$X_3' \leftarrow r - H^3 + R^2 - a_2 Z_3^2$
$X_3 \leftarrow X_3' - r$; $Y_3 \leftarrow -S_1 H^3 - X_3 R$
return (X_3, Y_3, Z_3)

ECDBL-D1-D3 needs 5 extra multiplications and 11 additions compared to ECDBL. ECADD-D1 needs 1 extra square and 3 additions, and 1 multiplication less compared to ECADD.

The formula below brings addition of points without any zero intermediate value.

Algorithm 4 ECADD-D1-A4

Input: $P = (0, Y_1, Z_1), Q = (X_2, Y_2, Z_2) \in E^{\mathcal{J}}$,
$r, s = r^2$
Output: $P + Q$
$H \leftarrow X_2 Z_1^2$
$S_1 \leftarrow Y_1 Z_2^3$; $S_2 \leftarrow Y_2 Z_1^3$
$R' \leftarrow r + S_2 - S_1$; $Z_3 \leftarrow Z_1 Z_2 H$
$R'' \leftarrow R' - 2r$
$X_3' \leftarrow r - H^3 + R' R'' - a_2 Z_3^2 - s$
$Y_3' \leftarrow r - S_1 H^3 - X_3 R' - r X_3$
$X_3 \leftarrow X_3' - r$; $Y_3 \leftarrow Y_3' - r$
return (X_3, Y_3, Z_3)

ECADD-D1-A4 needs 2 extra multiplications and 10 additions compared to ECADD.

For the Double-and-Add always (Algorithm 1), the extra cost of applying ECDBL-D1-D3 and ECADD-D1-A4 is $7n$ multiplications and $21n$ additions.

5 Dynamically check the curve and the base point

Modifying the formulæ, as described in the previous section, is expensive. Sometimes, the protection is not necessary because the elliptic curve does not contain any zero-value points. In this case, the protection brings extra unnecessary computation.

We give in this section our new method to save the extra costly field operations required for ECDBL-D1-D3 and ECADD-D1-A4 compared to ECDBL and ECADD. The given curve and the given point can be checked to remove some unnecessary protection.

For the analysis of the gain performance of our method, by convention, we set that the cost of a multiplication is equal to the cost of a square. α will denote the ratio of the cost

of an addition/subtraction to the cost of a multiplication. It is connected to the bit length of the manipulated integers and depends on the architecture. We can refer to the analysis made by Giraud and Verneuil in [8], which is given in Table 1.

Bit length	160	192	224	256	320	384	512	521
α	0.36	0.30	0.25	0.22	0.16	0.13	0.09	0.09

Table 1. Ratio of a cost of an addition/subtraction to the cost of a multiplication given in [8]

5.1 Legendre symbol

For our new protection, we need the definition and some properties of the Legendre symbol $\left(\frac{a}{p}\right)$.

Definition 4. *Let p be an odd prime, and a an integer.*

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is a square modulo } p \text{ and } a \not\equiv 0 \mod p, \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } p, \\ 0 & \text{if } a \equiv 0 \mod p \ . \end{cases}$$

The Legendre symbol can be computed using the generalized Jacobi symbol. The following algorithm permits to compute the Jacobi symbol.

Algorithm 5 Binary algorithm for the Jacobi symbol

Input: an odd integer $0 < b < 2^n$, and $0 < a < b$, with $gcd(a, b) = 1$
Output: $\left(\frac{a}{b}\right)$

```
J ← 1
α ← n                                   ▷ bit length of a
β ← n                                   ▷ bit length of b
while a ≠ 1 do
    while a is even do
        a ← a/2; α ← α − 1
        if (b = 3 mod 8 or b = 5 mod 8) then J = −J
    end while
    if α ≤ β then
        swap(a, b); swap(α, β)
        if (a = 3 mod 4 and b = 3 mod 4) then J = −J
    end if
    if ((a + b) = 0 mod 4) then a ← (a + b)/4; α ← α − 1
    else a ← (a + 3b)/4
end while
return J
```

The complexity of the algorithm is $\mathcal{O}(n^2)$. However, we performed statistical tests that reveal that the average number of additions for random values of a and random primes b is $1.5n$. We neglect shift, modulo 4 and modulo 8 operations.

5.2 Checking that the curve does not contain any zero x-coordinate

Given the elliptic curve $E\colon y^2 = x^3 + ax + b$, and from Proposition 1, we can state that

- if $\left(\frac{b}{p}\right) = 1$, then the curve might contain some zero x coordinate points,
- if $\left(\frac{b}{p}\right) = -1$, the curve does not contain any zero x coordinate point.

One can compute the Jacobi symbol of b at the beginning of the ECSM. If $\left(\frac{b}{p}\right) = 1$, then the protection against (D1) is applied. If $\left(\frac{b}{p}\right) = -1$, the protection against (D1) is not necessary. In this case, we can remove the protection and save 4 multiplications and 3 additions over ECDBL-D1-D3 and 1 multiplication and 3 additions over ECADD-D1-A4.

On random elliptic curves, the probability of b being a square is $1/2$. The Jacobi symbol costs $1.5n$ additions in average. On the Double-and-Add always method (Algorithm 1), the performance gain is $5/2n + 6/2\alpha n - 1.5\alpha n = (2.5 + 1.5\alpha)n$ multiplications.

5.3 Checking that the curve does not contain any zero tangent line slope points

This case is analogous to the previous subsection. Given the elliptic curve $E\colon y^2 = x^3+ax+b$, and from corollary 1, we can state that

- if $\left(\frac{-3a}{p}\right) = 1$, then the curve might contain some zero tangent line slope points,
- if $\left(\frac{-3a}{p}\right) = -1$, the curve does not contain any zero tangent line slope point.

One can compute the Jacobi symbol of $-3a$ at the beginning of the ECSM. If $\left(\frac{-3a}{p}\right) = 1$, then the protection against (D3) is applied. If $\left(\frac{-3a}{p}\right) = -1$, the protection against (D3) is not necessary. In this case, we can remove the protection and save 1 multiplication and 8 additions over ECDBL-D1-D3.

On random elliptic curves, the probability of $-3a$ being a square is $1/2$. The Jacobi symbol costs $1.5n$ additions in average. On the Double-and-Add always method (Algorithm 1), the performance gain is $1/2n + 8/2\alpha n - 1.5\alpha n = (0.5 + 2.5\alpha)n$ multiplications.

5.4 Checking that the base point is not a y same coordinate point

We are interested here on ECSMs where the base point (or its inverse) is involved at each iteration of the ECSM, like the Double-and-Add method and the Montgomery Ladder using co-Z formulæ [10, Algorithm 7].

Given the elliptic curve $E\colon y^2 = x^3 + ax + b$, the base point $P = (x_P, y_P)$ and from Proposition 3, we can state that

- if $\left(\frac{-3x_P^2-4a}{p}\right) = 1$, then P is a y same coordinate point,

– if $\left(\frac{-3x_P^2-4a}{p}\right) = -1$, P is not a y same coordinate point.

One can compute the Jacobi symbol of $-3x_P^2 - 4a$ at the beginning of the ECSM. If $\left(\frac{-3x_P^2-4a}{p}\right) = 1$, then the protection against (A4) is applied. If $\left(\frac{-3x_P^2-4a}{p}\right) = -1$, the protection against (A4) is not necessary when adding the base point. We can therefore remove the protection and save 1 multiplication and 7 additions over ECADD-D1-A4.

On random elliptic curves, the probability of the base point being a y same coordinate is $1/2$. The Jacobi symbol costs $1.5n$ additions in average. On the Double-and-Add always method (Algorithm 1), the performance gain is $1/2n + 7/2\alpha n - 1.5\alpha n = (0.5 + 2\alpha)n$ multiplications.

A more efficient method to prevent from y same coordinate points is described in the next subsection.

5.5 Modifying the base point

We propose another method to protect against (A4).

Given the elliptic curve $E\colon y^2 = x^3 + ax + b$ and the base point $P = (x_P, y_P)$, the Jacobi symbol of $\left(\frac{-3x_P^2-4a}{p}\right)$ is computed to check if P is a y same coordinate point. If it is not, the protection is not applied. If P is a y same coordinate point, rather than applying the protection, we propose the following method illustrated in Algorithm 6.

Algorithm 6 Protected ECSM against (A4)

Input: $P = (x_P, y_P) \in E$ and an integer k
Output: $[k]P$
  $j \leftarrow 0$
  $S = (x_S, y_S) \leftarrow P$
  while $\left(\frac{-3x_S^2-4a}{p}\right) = 1$ **do**
    $S \leftarrow 2S$
    $j \leftarrow j + 1$
  end while
  Compute $Q \leftarrow \left[\lfloor k/2^j \rfloor\right] S$ without protection against (A4)
  Compute $R \leftarrow Q + [k \mod 2^j]P$ with protection against (A4)
  return R

After the while loop, S is not a y same coordinate point. Therefore, the protection against (A4) is not necessary when computing Q.

After performing a point doubling, the point S is in affine coordinates. If $S = (X_S, Y_S, Z_S)$ is in Jacobian coordinates, it is equivalent to check $\left(\frac{-3X_S^2-4aZ_S^4}{p}\right)$. Indeed, $-3X_S^2 - 4aZ_S^4 = (-3x_S^2 - 4a)Z_S^4$ and Z_S^4 is a square. With the Legendre properties, $-3X_S^2 - 4aZ_S^4$ is a square if $-3x_S^2 - 4a$ is.

We performed a statistical study on standardized curves. Running the algorithm with random inputs, j is equal to 1 in average[8]. The point R is computed under protection against (A4), with a very small scalar (a very few bits). The computation is negligible compared to the complete ECSM. The Jacobi symbol costs $1.5n$ additions in average, which is computed 2 times in average. On the Double-and-Add always method (Algorithm 1), the performance gain is $n + 7\alpha n - 3\alpha n = (1 + 4\alpha)n$ multiplications which is better than the gain of the protection against (A4) of Section 5.4: $(0.5 + 2\alpha)n$ multiplications.

6 Conclusion

An analysis on the ZPA is given. We suggest a method to dynamically check the curve and the base point. Depending on the given curve and base point, the formulæ used to perform the ECSM are dynamically adapted for protection against zero value points. The unnecessary protections are removed for efficiency.

The countermeasure needs the computation of the Jacobi symbol. The performance gain of the proposed method is given with a basic software Jacobi symbol module. A dedicated embedded Jacobi symbol calculator can improve the countermeasure.

References

1. T. Akishita and T. Takagi, *Zero-Value Point Attacks on Elliptic Curve Cryptosystem*. Proceedings of ISC'03, LNCS vol. 2851, Springer-Verlag, 2003, pp. 218-233.
2. T. Akishita and T. Takagi, *On the Optimal Parameter Choice for Elliptic Curve Cryptosystems Using Isogeny*. Proceedings of PKC'04, LNCS vol. 2947, Springer-Verlag, 2004, pp. 346-359.
3. M. Ciet and M. Joye, *(Virtually) Free Randomization Techniques for Elliptic Curve Cryptography*. Proceedings of ICIS'03, LNCS vol. 2836, Springer-Verlag, 2003, pp. 348-359.
4. H. Cohen, A. Miyaji and T. Ono, *Efficient Elliptic Curve Exponentiation Using Mixed Coordinates*. Proceedings of ASIACRYPT'98, LNCS vol. 1514, Springer-Verlag, 1998, pp. 51-65.
5. J.-S. Coron, *Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems*. Proceedings of CHES'99, LNCS vol. 1717, Springer-Verlag, 1999, pp. 292-302.
6. C. Clavier and M. Joye, *Universal Exponentiation Algorithm*. Proceedings of CHES'01, LNCS vol. 2162, Springer-Verlag, 2001, pp. 300-308.
7. J.-L. Danger, S. Guilley, P. Hoogvorst, C. Murdica and D. Naccache, *Low-Cost Countermeasure against RPA*. Proceedings of CARDIS'12, LNCS vol. 7771, Springer-Verlag, 2013, pp. 106-122.
8. C. Giraud and V. Verneuil, *Atomicity Improvement for Elliptic Curve Scalar Multiplication*. Proceedings of CARDIS'10, LNCS vol. 6035, Springer-Verlag, 2010, pp. 80-101.
9. L. Goubin, *A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems*. Proceedings of PKC'03, LNCS vol. 2567, Springer-Verlag, 2002, pp. 199-210.
10. R. R. Goundar, M. Joye and A. Miyaji, *Co-Z Addition Formulæ and Binary Ladders on Elliptic Curves - (Extended Abstract)*. Proceedings of CHES'10, LNCS vol. 6225, Springer-Verlag, 2010, pp. 65-79.
11. K. Itoh, T. Izu and M. Takenaka, *Efficient Countermeasures against Power Analysis for Elliptic Curve Cryptosystems*. CARDIS'04, Klumer, 2004, pp. 99-114.
12. M. Joye and C. Tymen, *Protections against Differential Analysis for Elliptic Curve Cryptography*. Proceedings of CHES'01, LNCS vol. 2162, Springer-Verlag, 2001, pp. 377-390.
13. N. Meloni, *New Point Addition Formulae for ECC Applications*. Proceedings of WAIFI'07, LNCS vol. 4547, Springer-Verlag, 2007, pp. 189-201.
14. N. Smart, *An Analysis of Goubin's Refined Power Analysis Attack*. Proceedings of CHES'03, LNCS vol. 2779, Springer-Verlag, 2003, pp. 281-290.
15. E. Trichina and A. Bellezza, *Implementation of Elliptic Curve Cryptography with Built-In Counter Measures against Side Channel Attacks*. Proceedings of CHES'02, LNCS vol. 2523, Springer-Verlag, 2002, pp. 98-113.

[8] This is related to the probability that a point is a y same value point is $1/2$

16. L. Washington *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman & Hall/CRC, 2008.

A Division Polynomials

Let $E\colon y^2 = x^3 + ax + b$ defined over $\mathbb{F}_p$. The division polynomials $\psi_m \in \mathbb{F}_p[x, y]$ are defined as:

$$\begin{aligned}
\psi_0 &= 0 \\
\psi_1 &= 1 \\
\psi_2 &= 2y \\
\psi_3 &= 3x^4 + 6ax^2 + 12bx - a^2 \\
\psi_4 &= 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3 \\
\psi_{2m+1} &= \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 \text{ for } m \geq 2 \\
\psi_{2m} &= (2y)^{-1}\psi_m(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2) \text{ for } m \geq 3 \ .
\end{aligned}$$