Same Value Analysis on Edwards Curves

Rodrigo Abarzúa[1], Santi Martínez[2] and Valeria Mendoza[1]

[1]Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Chile.
[1]{rodrigo.abarzua, valeria.mendoza}@usach.cl.
[2]Departament de Matemàtica, Universitat de Lleida, Spain.,
[2]santi@matematica.udl.cat

July 21, 2015

Abstract

Recently, several research groups in cryptography have presented new elliptic curve model based on Edwards curves.

These new curves were selected for their good performance and security perspectives.

Cryptosystems based on elliptic curves in embedded devices can be vulnerable to Side-Channel Attacks (SCA), such as the Simple Power Analysis (SPA) or the Differential Power Analysis (DPA).

In this paper, we analyze the existence of special points whose use in SCA is known as Same Value Analysis (SVA), for Edwards curves. These special points show up as internal collisions under power analysis. Our results indicate that no Edwards curve is safe from such an attacks.

Keywords: Elliptic curve cryptography, Side-channel attack, Same value analysis, Edwards curves, Smart cards.

1 Introduction

The demand on wireless technology (cell phones, smart cards, wireless sensor networks etc.) has been increasing in the last years. Most of these devices rely on embedded microprocessors to securely transmit data. Designing efficient cryptographic algorithms is a fundamental problem in the development of secure wireless devices.

Embedded devices present specific problems for implementation of cryptosystems. On the one hand, they are computationally limited. On the other hand, they are susceptible to attacks that do not concern conventional cryptography.

Elliptic Curve Cryptography (ECC) is a class of public-key cryptosystems proposed by Miller [30] and Koblitz [25], that addresses the first problem, by providing significant advantages in several situations, since smaller keys can be used without decreasing the security level. For example, some industry standards require 2048 bits for the size of integers [1] in the RSA system, whereas the equivalent requirement for elliptic curve cryptography (ECC) is to work with finite fields of 224 bits [20]

In this way, the computations are faster, therefore not so much computing power is needed. Moreover, there is a great interest in the industry to improve the efficiency for the scalar multiplication of points on the elliptic curve, since it is one of the most time consuming operations in ECC protocols [18].

With regard to the second problem, ECC on Smart-Cards can be vulnerable to *Passive Side-Channel Attacks.* This attack exploit physical leakages from cryptographic process executing on a device, for example: timing [26], power consumption [27] and electromagnetic radiation [19, 34].

There are two general strategies for these attacks: *Simple Side-Channel Analysis* (SSCA) [26], which analyzes the measurements obtained during a single scalar multiplication, based on the variations observed in the measurements due to the bit value of the secret key; and *Differential Side-channel Analysis* (DSCA) [27], which is based on statistical techniques to retrieve information about the secret key based on measurements from several scalar multiplications.

In 2003 Goubin in [21] presented RPA. The basic idea of this attack is to used special points P_0 on the elliptic curve $E(\mathbb{K})$ such that one of the coordinates is 0 in $\mathbb{K}$. An attacker could construct a point from the elliptic curve used in the embedded devices, so that depending on the assumed value of a specific bit of the key, she should obtain a special point during the scalar multiplication. An operation which includes a special point is faster and less power consuming than usual. Thus, the attacker can notice when a special point appears and, after several executions, she could obtain the secret key.

Akishita and Takagi in [1] presented a generalization of Goubin's attacks. An attacker can obtain information not only with the special points, but also when auxiliary registers take the value zero. These new attacks were called Zero Value Point attacks (ZVP-attacks). Akishita and Takagi gave some conditions that, under which, the curve can be attacked through ZVP-attacks.

Murdica *et al.* in [32] found another kind of attack in which the attacker can force some points to have collisions, that is to say, they produce some identical values (and identical side-channel traces) during the encryption. This way, she can obtain information about the secret, and after some iterations, she can know the whole secret value. This attack is called Same Value Analysis (SVA).

On the other hand, several groups have presented new elliptic curves for model of Edwards that offer good performance and provide a stronger overall security [3, 5, 8, 9, 22, 28]. Martínez *et al.* presented an analysis of ZVP points to the Edwards curves, in which they show that these curves are secure against ZVP-attacks [29].

The aim of this paper is to study SVA points for Edwards models. Our results show us that elliptic curves that recently submitted by community, have points that enable SVA-type attacks.

This paper is organized as follows. Section 2 provides an introduction to Edwards curves. Section 3 introduces the SVA points on Edwards and twisted Edwards curves. In Section 4 we presents our experimental results on the existence of SVA points in the new curves Edwards.

[1] For the modulus (n).

Finally, Section 5 concludes the paper.

2 Edwards Curves

Edwards, [17] introduced a new form for elliptic curve:

$$x^2 + y^2 = c^2(1 + x^2y^2),$$

Every elliptic curve over a non-binary finite field can be birationally transformed to an Edwards curve. Some elliptic curves require moving to an extension field for the transformation, but others have transformations defined on the original finite field.

Bernstein and Lange [6] extended the result of Edwards to capture a larger class of elliptic curves over the original field, and introduced their use for cryptographic applications. Their model is

$$E : x^2 + y^2 = c^2(1 + dx^2y^2),$$

where $cd(1 - c^4d) \neq 0$. The only singular points for these curves is the point at infinity. The neutral element of the curve is $(0, c)$ and given a pair of points $(x_1, y_1), (x_2, y_2) \in E$ the addition law is:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{c(1 + dx_1x_2y_1y_2)}, \frac{y_1y_2 - x_1x_2}{c(1 - dx_1x_2y_1y_2)} \right)$$

When d is not a square, the Edwards addition law is complete: it is defined for all pairs of input points on the Edwards curves. Moreover, this addition law unified, it works for every pair of points, making no distinction between addition, doubling, operations whit the neutral element and symmetries.

Given that inversions in the fields have a high computational cost, Bernstein and Lange [6] also provided a (homogenized) projective coordinates model $(X^2+Y^2)Z^2 = c^2(Z^4+dX^2Y^2)$. In which, the point $(X_1 : Y_1 : Z_1), Z_1 \neq 0$ corresponds to the affine point $(X_1/Z_1, Y_1/Z_1)$. The neutral element is $(0 : c : 1)$ and the symmetric element of $(X_1 : Y_1 : Z_1)$ is $(-X_1 : Y_1 : Z_1)$

2.1 Addition in projective coordinates

Given two points $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$ an Edwards elliptic curve, the addition $(X_3 : Y_3 : Z_3) = (X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$ is computed as:

$$\begin{aligned}
&A = Z_1Z_2, \quad B = A^2, \quad C = X_1X_2, \quad D = Y_1Y_2, \\
&E = dCD, \quad F = B - E, \quad G = B + E, \\
&X_3 = AF((X_1 + Y_1)(X_2 + Y_2) - C - D), \\
&Y_3 = AG(D - C), \\
&Z_3 = cFG.
\end{aligned}$$

2.2 Doubling in projective coordinates

Given a point $(X_1 : Y_1 : Z_1)$ on the Edwards curve, the doubling $(X_3 : Y_3 : Z_3) = 2(X_1 : Y_1 : Z_1)$, is computed as:

$$
\begin{aligned}
&B = (X_1 + Y_1)^2, \quad C = X_1^2, \quad D = Y_1^2, \\
&E = C + D, \quad H = (cZ_1)^2, \quad J = E - 2H, \\
&X_3 = c(B - E)J, \\
&Y_3 = cE(C - D), \\
&Z_3 = EJ.
\end{aligned}
$$

3 Special Points on Elliptic Curves and Countermeasures

Implementations of ECC may be vulnerable to physical attacks unless special countermeasures are implemented. These attacks are called Side-Channel Attacks (SCA)[23].

There are several methods to obtain the secret, depending on how many executions are carried out and what kind of information is used. There are three main types of attacks: power, time and electromagnetic attacks (EMA). All attacks can also be divided into *Simple Power Analysis* (SPA) and *Differential Power Analysis* (DPA).

SPA looks for patterns of computation behavior from the information obtained from a single execution. With this information, the attacker tries to guess which operations are being carried out by the embedded devices at every moment, and from this guess the secret used in the cryptosystem. DPA obtains information after several executions. It computes statistical analysis from the information obtained and looks for the most probable key.

In order to obtain the key using timing attacks, the secret must be related to the time needed to process different inputs. In other words, to find variations in the obtained measurements in a given moment in order to obtain the bit that is processed in that moment. If we can obtain the bit at specific steps of the scalar multiplication, after many executions we can obtain the whole secret.

EMA attacks are possible because embedded devices are made of silicon semi-conductors. When the embedded devices is accessed, there is a flow of electrons in the medium, which generates electromagnetic waves. These waves can be captured by an attacker, obtaining information about the operations carried out and giving a chance to guess the secret.

3.1 Refined Power-Analysis (RPA) Attacks

Under some conditions, an attacker can construct points such that after a specific (chosen) sequence of doubling and addition computations, she obtains a point with abscissa or ordinate equal to zero. In this way, after some executions, she could obtain the secret, k in $[k]P$. Goubin [21] called these points, "special points" and established the conditions to decide if such points exist in a given elliptic curve. The conditions that must avoid an elliptic curve in Weierstrass form should avoid in order that it does not contain any special points are:

- b is a quadratic residue in $\mathbb{F}_p$. This way, the curve will have $(0, y)$ points.
- the elliptic curve has points of order 2, which are $(x, 0)$ points.

Smart [37] proposed the use of isogenies in order to avoid RPA, since the fact that a given elliptic curve fulfills the Goubin's conditions does not mean that its isogenous curves have the same problem.

3.2 Zero-Value Point (ZVP) Attacks

An attacker can also obtain information about the secret if there exists some intermediate parameters, used in the addition or doubling computations, with zero-value. This attack is called Zero-Value Point attack (ZVP-attack).

When an embedded devices is computing, an attacker can know if there is a zero-value involved since the power consumption is lower than usual.

Akishita and Takagi [1, 2] gave conditions for elliptic curves to be vulnerable to this kind of attacks:

- ED1. $3x^2 + a = 0$,
- ED2. $5x^4 + 2ax^2 - 4bx + a^2 = 0$,
- ED3. $\text{order}(P) = 3$,
- ED4. $x(P) = 0$ or $x(2P) = 0$, then b is a quadratic residue,
- ED5. $y(P) = 0$ or $y(2P) = 0$.

Conditions ED4 and ED5 are those proposed by Goubin.

3.3 Same Value Analysis (SVA) Attacks

Murdica *et al.* [32] found another kind of attack in which the attacker can force some points to have collisions, that is to say, they produce some identical values during the encryption. By identifying these values, she can obtain information about the secret, and after some iterations she can know the whole secret value. This attack is called Same Value Analysis (SVA). Murdica, notes that certain special points, internal collisions occur in the operation of doubling Jacobian coordinates, for example, SED conditions come from the doubling of a point $P = (x, y)$:

- SED2. $x = 0$,
- SED3. $x = 1$,
- SED15. $2y = 3x^2 + a$.

3.4 Recursive attack using special points

Suppose that the attacker already knows the $l - i - 1$ left-most bits of the fixed scalar $k = (k_{l-1}, k_{l-2}, \ldots, k_{i+1}, k_i, k_{i-1}, \ldots, k_0)_2$ and wants to recover the unknown bit k_i.

The attacker chooses a point P_0 satisfying one of the above conditions (SED2 for example), computes the point

$$P = [(k_{l-1}, k_{l-2}, \ldots, k_{i+1}, k_i, k_{i-1}, \ldots, k_0)_2^{-1} \bmod \#E]P_0$$

and sends this P to the targeted chip that computes the ECSM using the fixed scalar k. If $k_i = 0$, the point P_0 will be doubled during the ECSM, and a collision of power consumption will appear during the ECSM. The attacker recovers several traces of the power consumption during the computation of $[k]P$. Using the methodology presented in [35] and [13] to detect internal collision. He can conclude if $k_i = 0$ or not ($k_i = 1$) The attacker can recursively recover all bits of k.

3.5 Countermeasures

In this section we present the main countermeasures for RPA, ZVP, and SVA. There are three major families, *a*) Isogeny, *b*) Randomizing scalar k and *c*) Randomizing point P.

3.5.1 Isogeny Countermeasure

In order to resist RPA for point of large order, Smart [37] proposed to map the underlying curve to the isogenous curve that does not have the point $(0, y)$. To apply the isogeny defense it would be better to alter the standards so that the curves are replaced with isogenous ones. However, since this is unlikely to be an option the smart card needs to convert the input point to the isogenous curve. This countermeasure with a small isogeny degree is faster. However, but it is much slower to implement on a smart card, in [4]. The use of isogenies is more efficient as the same algorithms can be used for point arithmetic with the addition of two transformations. Moreover, these transformations can be defined when a cryptosystem is implemented to minimize the impact on the time taken to compute a scalar multiplication.

Volcanoes Isogeny [31]: Miret *et al.*[31], presented improvements the computing and time to find the suitable isogeny degrees of several SECG standard curves [36]. Miret *et al.* uses the Isogeny-Volcano. Moreover, evaluated the probability of a curve failing for this condition (ED4: $x(P) = 0$ or $x(2P) = 0$, i.e. b is a quadratic residue) is $1/2$.

In Table 1 we present the result obtained by Smart in [37] and Mired *et al.* in [31]. In the second and fourth column present the minimal and preferred isogeny degree with respect to condition ED4, given by Smart in [37], while the third and the fifth columns contain the degrees of the isogeny-route given by the algorithm presented by Miret in [31]. More precisely, the minimal $\ell_{\mathbf{std}}$ and preferred isogeny degree $\ell_{\mathbf{prf}}$ can be defined:

$\ell_{\mathbf{std}}$: The minimal Isogeny degree for condition ED4.

$\ell_{\mathbf{prf}}$: The minimal Isogeny degree for condition ED4 and condition $a = -3$.

Table 1: Minimal isogeny degrees with respect to ED4 for SECG curves

ED4	$\ell_{\mathbf{std}}$ [37]	$\ell_{\mathbf{std}} - route$ [31]	$\ell_{\mathbf{prf}}$[37]	$\ell_{\mathbf{prf}}$-route [31]
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

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 [31]. For example, the curve P-192 from NIST the preferred isogeny degree is $\ell_{\mathbf{prf}} = 73$ while $\ell_{\mathbf{prf}} - route = 5 - 13 - 23$, which means the composition of three isogenies of degrees 5, 13 and 23.

In Table 2 provides the computing and searching time to find the suitable isogeny degrees, as can be seen the volcano-isogeny obtain better calculation and search times, specially when searching for a preferred curve.

Table 2: Time for computing minimal isogeny degrees with respect to ED4

T. Comp./T.Cerca.	Methods anterior (seg.)	Isogeny-route (seg.)
P-192 (l_{std})	44.30/51.24	6.01/ 6.99
P-192 (l_{prf})	3474.7/4788.15	50.31/59.22
P-256 (l_{prf})	5.93/6.43	0.035/0.043

To protect against RPA and ZPA attacks, the base point P or the secret scalar k should be randomized.

3.5.2 Countermeasures Randomization of the Scalar k

Scalar Randomization [15]: Select a random number d and compute the scalar multiplication $Q = [k']P = [k + d(\#E)]P = [k]P + [d(\#E)]P = [k]P$, since $[d(\#E)]P = P_\infty$. Since, the d of size 20-bits. We have considered the average of percentage of performance loss of curves P$-$192,

Table 3: Theoretical loss Cost

	n bits of d		
NIST curves	20-bits	32-bits	40-bits
P$-$192	10.4%	16.6%	20.8%

Exponent Splitting [14]: For any random number r is a $n-$bit random integer, that is, of the same bit length as k, and computing $[k]P = [k-r]P+[r]P$.

Generate a random number r is expensive, this countermeasure requires at least twice the processing both $[k-r]P$ and $[r]P$ need to be computed[2].

Trichina-Bellezza Countermeasure [40]: Trichina *et al.* was proposed the following countermeasure. For any random number r, evaluate $[k]P = [kr^{-1}]([r]P)$. A disadvantage of this countermeasure is required to compute the inverse of r module $\mathrm{ord}_E(P)$. Besides, two scalar multiplication are needed, first $R = [r]P$ is computed, then $[kr^{-1}]R$ is computed.

Euclidean Division [12]: 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 $n/2$. Ciet, presented an regular algorithm variant of Shamir's double ladder.

Self-Randomized Exponentiation Algorithms [11]: 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 exponent k. Defining

$$k_{d\to j} := (k_d, \ldots, k_j)_2 = \sum_{j \le i \le d} k_i 2^{i-j},$$

The main idea behind *self-randomized exponentiation* consists in taking part of k as a source of randomness. The algorithm relies on the simple observation that, for any $0 \le i_j \le l$, we have.

$$\begin{aligned}
[k]P &= [k_{l\to 0}]P \\
&= [k_{l\to 0} - k_{l\to i_1}]P + [k_{l\to i_1}]P \\
&= [(k_{l\to 0} - k_{l\to i_1}) - k_{l\to i_2}]P + [k_{l\to i_1}]P + [k_{l\to i_2}]P \\
&= \vdots \qquad\qquad\qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \\
&= [(((k_{l\to 0} - k_{l\to i_1}) - k_{l\to i_2}) - k_{l\to i_3}) \cdots - k_{i_f}]P + [k_{l\to i_1}]P \\
&\quad + [k_{l\to i_2}]P + [k_{l\to i_3}]P + \cdots + [k_{l\to i_f}]P\ .
\end{aligned}$$

3.5.3 Set of Countermeasures Randomization Point P

Blinding Point [15]: A fixed (secret) point R is selected and $S = [k]R$ is precomputed. Given P, the computation of $[k]P$ is replaced by that of $[k](P+R)$ and the known value $S = [k]R$ is subtracted at the end of the computation.

3.5.4 Countermeasures do not support special points attacks:

Randomized algorithm $2P^*$ [12]: Ciet and Joye in [12], proposed the 2^* Algorithm, this randomization method is applicable to left-to-right scalar multiplication algorithm. The idea is randomize $[2]P$ using the method of randomized projective coordinates. This allows to keep using P in affine coordinate (which

[2]Ebeid in [16], studied implementations of this countermeasure using the algorithm of Shamir-Strauss method [39], and finds a vulnerability which can be attacked by a DPA (Lemma 6.1), its study recommends, for this countermeasure each term of $[k-r]P$ and $[r]P$ should be computed separately using a SPA-resistant algorithm.

equivalent to saying that its Z-coordinate of P is equal to 1) and the scalar multiplication is computed in mixed coordinates is used (which is more efficient, then using only projective coordinates). Moreover, the algorithm allows using an elliptic curve with parameter $a = -3$.

Randomized field $\mathbb{K}$ Isomorphism [24]: The idea of this countermeasure is to use a representation of random fields definition of elliptic curve, i.e. use a Randomized field $\phi : \mathbb{K} \to \mathbb{K}'$ then this isomorphism is used to obtain a point $P' = \phi(P)$ of the curve $E' = \phi(E)$, so the scalar multiplication is calculated as

$$[k]P = \phi^{-1}([k](\phi(P)))$$

This countermeasure has the major disadvantage that special fields used in most standards for example NIST and SEGC use irreducible polynomials for which the reduction is much more efficient (for example, in characteristic two using trinomials or pentanomials in which most of the terms have very low degree), this property is usually lost after the field isomorphism is applied, so the operations over the isomorphic fields can be much slower (see [4]).

Randomized $E(\mathbb{K})$ Isomorphism [24] The idea of this countermeasure is to transfer the base point $P_1 = (x, y) \in E_1(\mathbb{K})$ randomly selected isomorphic curve $\phi : E_1(\mathbb{K}) \to E_2(\mathbb{K})$ (the parameters of the curve $E_2(\mathbb{K})$ are $a' = r^4a$ and $b' = r^6b$), the transferred point is $\phi(P_1) = (r^2x, r^3y) = P_2$ and the scalar multiplication is execute as $([k]P_2 = [k]\phi(P_1))$ on the curve $E_2(\mathbb{K})$ and the result $Q_2 = (x_k, y_k)$ is brought back to the original curve $E_1(\mathbb{K})$, by computing $Q_1 = [k]P = (x_k/r^2, y_k/r^3) = \phi^{-1}([k](\phi(P)))$. The randomization takes $4M + 2S$ at the beginning and $1I + 3M + 1S$ at the end. However, when using random curve isomorphisms the parameters of $E_2(\mathbb{K})$ cannot be chosen and one cannot take advantage of algorithms that require curve parameters to be set to specific values, the curve parameter a is randomized. In particular, fast doubling formula for $a = -3$ cannot be used.

Generalization: Tunstall and Joye, [41]: 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 $P = \phi^{-1}(P') = (f^{\nu}X', f^{\mu}Y', f^{\mu+\nu}Z)$. The case $\mu = 2$, $\nu = 3$ correspond to the technique of randomized $E(\mathbb{K})$ isomorphism of Joye and Tymen [24].

Randomized Projective Coordinates [15]: Randomizing the homogeneous projective coordinates of point $P = (X, Y, Z)$ with $\lambda \neq 0$ to $P = (\lambda X, \lambda Y, \lambda Z)$. The random variable λ can be updated in every execution or after each doubling or addition. When we compute the scalar multiplication using Jacobian coordinates, the point Q is represented as $Q = (X, Y, Z)$. The point Q must be recovered to affine coordinate by computing $x = X/Z^2$ and $y = Y/Z^3$ and so we avoid the attack presented in [33]. Moreover, when using Jacobian coordinates it is suggested to selected curve parameters a as $a = -3$. Using randomized projective coordinates is much more efficient but does not allow λ to be set to one [38], i.e. scalar multiplication we cannot use mixed coordinates (Jacobian or affine). This countermeasure is effective against Template Attacks [10].

In the following table we present a summary of countermeasures to DPA, each with its overhead cost and whether it is effective (✓) or fails (×) against the three types of attacks considered: RPA, ZVP and SVA.

Table 4: Summary Countermeasure

Countermeasure	**Computation Overhead**[3]	**RPA**	**ZVP**	**SVA**
Isogeny Defense [37]	Negligible	✓	✓	✓
Scalar Randomization [15]	Low[4]	✓	✓	✓
Exp. Splitting [14]	High[5]	✓	✓	✓
Trichina-Bellezza [40]	High[6]	✓	✓	✓
Euclidean Division [12]	Medium[7]	✓	✓	✓
Self-Randomized Expo. [11]	Negligible[8]	✓	✓	✓
Blinding the Point [15]	High[9]	✓	✓	✓
Randomized Projective [15]	Negligible[10]	×	×	×
Method $2P^*$ [12]	Negligible	×	×	×
Random Field $\mathbb{K}$ Isomorphism [24]	Medium[11]	×	×	×
Random $E(\mathbb{K})$ Isomorphism [24]	Low[12]	×	×	✓

In the next session, we present the conditions for the existence of SVA points in the Edwards model, these conditions come from the algorithms for point addition and doubling.

3.6 SVA on Edwards Curves

Given the points $P_1 = (\lambda_1 x_1, \lambda_1 y_1, \lambda_1)$ and $P_2 = (\lambda_2 x_2, \lambda_2 y_2, \lambda_2)$, in projective coordinates on the Edwards curves E, we now show that the degrees in λ_1 and λ_2 of the terms computed during the doubling of P_1 or computed during the addition of P_1 and P_2 can be used to mount a SVA attacks.

Algorithm 3.6.1 and 3.6.2 give the addition and doubling formulas for Edwards curves in projective coordinates. Pairs of terms with matching degrees in λ_1 and λ_2 are those around which SVA points can be constructed since the occurrence of a collision between such terms does not depend on the value of λ_1 and λ_2.

[3] High: $\approx 100\%$, Medium: $(30-70)\%$, Low: $(10-25)\%$, Negligible: $< 0.5\%$

[4] On average performance loss is 15.9% for the curve P−192.

[5] 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[16]).

[6] Two scalar multiplication are needed.

[7] Using a regular algorithm variant of Shamir's double ladder, the algorithm presented by Ciet in [12] cost is $\frac{n}{2}D + \frac{n}{2}A$.

[8] Performance loss is $10A$ for a curve to P−192, for details see Alg II in [11].

[9] This countermeasure is considered inefficient, since it must perform two scalar multiplications $S = [k]p$ and $[k](P+R)$.

[10] 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.

[11] Mersenne or "sparse" primes cannot be used.

[12] $a = -3$ cannot be used.

Informations on the right show the degrees of the parameters λ_1, λ_2 of each operand and the result. In the doubling Algorithm, we denote by n an operand with a term λ of degree n. In the addition Algorithm, we denote $l_1 m_2$ an operand with a term λ_1 of degree l, and a term λ_2 of degree m.

3.6.1 Addition

Considering the formulas given in the previous section, we look for conditions so that the points involved in the addition formula can be used to mount an SVA.

Algorithm 1 Addition on Edwards Curves

Inputs: $P = (\lambda_1 x_1, \lambda_1 y_1, \lambda_1)$ y $Q = (\lambda_2 x_2, \lambda_2 y_2, \lambda_2)$

$A = Z_1 \cdot Z_2 = \lambda_1\lambda_2$	$(1_1 1_2 \leftarrow 1_1 \cdot 1_2)$
$B = A^2 = \lambda_1^2\lambda_2^2$	$(2_1 2_2 \leftarrow 1_1 1_2 \cdot 1_1 1_2)$
$C = X_1 \cdot X_2 = \lambda_1\lambda_2 x_1 x_2$	$(1_1 1_2 \leftarrow 1_1 \cdot 1_2)$
$D = Y_1 \cdot Y_2 = \lambda_1\lambda_2 y_1 y_2$	$(1_1 1_2 \leftarrow 1_1 \cdot 1_2)$
$E = dC \cdot D = \lambda_1^2\lambda_2^2(dx_1x_2y_1y_2)$	$(2_1 2_2 \leftarrow 1_1 1_2 \cdot 1_1 1_2)$
$F = B - E = \lambda_1^2\lambda_2^2(1 - dx_1x_2y_1y_2)$	$(2_1 2_2 \leftarrow 2_1 2_2 - 2_1 2_2)$
$G = B + E = \lambda_1^2\lambda_2^2(1 + dx_1x_2y_1y_2)$	$(2_1 2_2 \leftarrow 2_1 2_2 + 2_1 2_2)$
$H = (X_1 + Y_1) \cdot (X_2 + Y_2) - (C + D)$	
$= \lambda_1\lambda_2[(x_1 + y_1)(x_2 + y_2) - (x_1x_2 + y_1y_2)]$	$(1_1 1_2 \leftarrow 1_1 1_2 - 1_1 1_2)$
$I = A \cdot F = \lambda_1^3\lambda_2^3(1 - dx_1x_2y_1y_2)$	$(3_1 3_2 \leftarrow 1_1 1_2 \cdot 2_1 2_2)$
$X_3 = I \cdot H = \lambda_1^4\lambda_2^4(1 - dx_1x_2y_1y_2)\cdot$	
$[(x_1 + y_1)(x_2 + y_2) - (x_1x_2 + y_1y_2)]$	$(4_1 4_2 \leftarrow 3_1 3_2 \cdot 1_1 1_2)$
$J = D - C = \lambda_1\lambda_2(y_1y_2 - x_1x_2)$	$(1_1 1_2 \leftarrow 1_1 1_2 - 1_1 1_2)$
$K = A \cdot G = \lambda_1^3\lambda_2^3(1 + dx_1x_2y_1y_2)$	$(3_1 3_2 \leftarrow 1_1 1_2 \cdot 2_1 2_2)$
$Y_3 = K \cdot J = \lambda_1^4\lambda_2^4(1 + dx_1x_2y_1y_2)(y_1y_2 - x_1x_2)$	$(4_1 4_2 \leftarrow 3_1 3_2 \cdot 1_1 1_2)$
$Z_3 = cF \cdot G = \lambda_1^4\lambda_2^4 c[1 - (dx_1x_2y_1y_2)^2]$	$(4_1 4_2 \leftarrow 2_1 2_2 \cdot 2_1 2_2)$

In summary we have

Degree of $\lambda_1\lambda_2$	Terms
$1_1 1_2$:	$\{A, C, D, H, J\}$
$2_2 2_2$:	$\{B, E, F, G\}$
$3_1 3_2$:	$\{I, J\}$
$4_1 4_2$:	$\{X_3, Y_3, Z_3\}$

Where internal collisions occur in the following cases:

$$
\begin{aligned}
x_1x_2 &= 1 && \Rightarrow A = C, & x_1y_2 + y_1x_2 &= 1 && \Rightarrow A = H,\\
y_1y_2 &= 1 && \Rightarrow A = D, & x_1x_2 &= 0 && \Rightarrow D = J,\\
2x_1x_2 &= y_1y_2 && \Rightarrow C = J, & (x_1+y_1)(x_2+y_2) &= 2y_1y_2 && \Rightarrow H = J,\\
1 &= y_1y_2 - x_1x_2 && \Rightarrow A = J, & dx_1x_2y_1y_2 &= 1 && \Rightarrow B = E,\\
x_1x_2 &= y_1y_2 && \Rightarrow C = D, & 2dx_1x_2y_1y_2 &= 0 && \Rightarrow I = K,\\
x_1x_2 &= x_1y_2 + y_1x_2 && \Rightarrow C = H, & 2dx_1x_2y_1y_2 &= 1 && \Rightarrow E = F,\\
y_1y_2 &= x_1y_2 + y_1x_2 && \Rightarrow D = H, & x_1x_2y_1y_2 &= 0 && \Rightarrow B = F = G,
\end{aligned}
$$

$$
\begin{aligned}
dx_1x_2y_1y_2 &= 1 && \wedge && x_1y_2 + y_1x_2 + 1 + dx_1x_2y_1y_2 &&= 0 && \Rightarrow && X_3 = Z_3,\\
dx_1x_2y_1y_2 &= -1 && \wedge && y_1y_2 - x_1x_2 - c(1 - dx_1x_2y_1y_2) &&= 0 && \Rightarrow && Y_3 = Z_.3
\end{aligned}
$$

3.6.2 Doubling

Similarly, the expressions used also provide for doubling point $(X_3 : Y_3 : Z_3) = 2(X_1 : Y_1 : Z_1)$, which are:

Algorithm 2 Doubling on Edwards curves

Inputs: $P = (X_1, Y_1, Z_1) = (\lambda x_1, \lambda y_1, \lambda)$

1: $B = (X_1 + Y_1)^2 = \lambda^2(x_1 + y_1)^2$ $(2 \leftarrow 1)$
2: $C = X_1^2 = \lambda^2 x_1^2$ $(2 \leftarrow 1)$
3: $D = Y_1^2 = \lambda^2 y_1^2$ $(2 \leftarrow 1)$
4: $E = C + D = \lambda^2(x_1^2 + y_1^2)$ $(2 \leftarrow 2 + 2)$
5: $H = c(Z_1^2) = c\lambda^2$ $(2 \leftarrow 1)$
6: $J = E - 2H = \lambda^2[(x_1^2 + y_1^2) - 2c]$ $(2 \leftarrow 2 - 2)$
7: $K = B - E = \lambda^2[(x_1 + y_1)^2 - (x_1^2 + y_1^2)]$ $(2 \leftarrow 2 - 2)$
8: $L = C - D = \lambda^2(x_1^2 - y_1^2)$ $(2 \leftarrow 2 - 2)$
9: $X_3 = cK \cdot J = c\lambda^4[(x_1^2 + y_1)^2 - (x_1^2 + y_1^2)][(x_1^2 + y_1^2) - 2c]$ $(4 \leftarrow 2 \cdot 2)$
10: $Y_3 = cE \cdot L = c\lambda^4(x_1^4 - y_1^4)$ $(4 \leftarrow 2 \cdot 2)$
11: $Z_3 = E \cdot J = \lambda^4[(x_1^2 + y_1^2) - 2c][(x_1^2 + y_1^2) - 2c]$ $(4 \leftarrow 2 \cdot 2)$

In summary we have:

Degree of λ	Terms
2	$\{B, C, D, E, H, J, K, L\}$
4	$\{X_3, Y_3, Z_3\}$

Conditions where SVA points are presented, for doubling are:

$$\begin{aligned}
y_1 &= 0 && \Rightarrow B = C, B = E, & y_1^2 &= c && \Rightarrow D = H,\\
2x_1 + y_1 &= 0 && \Rightarrow B = C, & x_1^2 &= 2c^2 && \Rightarrow D = J,\\
x_1 &= 0 && \Rightarrow B = D, B = E, D = E, & x_1^2 + y_1^2 &= c^2 && \Rightarrow E = H,\\
x_1 + 2y_1 &= 0 && \Rightarrow B = D, & 2c^2 &= 0 && \Rightarrow E = J,\\
(x_1 + y_1)^2 &= c^2 && \Rightarrow B = H, & x_1^2 + y_1^2 &= 3c^2 && \Rightarrow H = J,\\
x_1 y_1 + c^2 &= 0 && \Rightarrow B = J, & x_1^2 &= c^2 && \Rightarrow C = H,\\
x_1^2 &= y_1^2 && \Rightarrow C = D \Rightarrow L = Y_3, & x_1^2 + y_1^2 &= 0 && \Rightarrow Y_3 = Z_3,\\
y_1^2 &= 2c^2 && \Rightarrow C = J, & y_1^2 + c^2 &= 0 && \Rightarrow Y_3 = Z_3\\
x_1 = 0 \wedge y_1 &= 0 && \Rightarrow K = X_3 = 0.
\end{aligned}$$

$$\begin{aligned}
c(2x_1y_1(x_1^2 + y_1^2 - 2c^2) - (x_1^4 - y_1^4)) &= 0 && \Rightarrow X_3 = Y_3,\\
(x_1^2 + y_1^2 - 2c^2)(2x_1y_1 - x_1^2 - y_1^2) &= 0 && \Rightarrow X_3 = Z_3,
\end{aligned}$$

3.6.3 Twisted Edwards Curve

Bernstein *et al.* in [7] introduced a generalization of Edwards curves.

Let $\mathbb{F}_q$ be a non binary field. Then, the *twisted Edwards curve* with coefficients $a, d \in \mathbb{F}_q$ satisfying $ad(a - d) \neq 0$, is a curve of the form

$$E_{E,a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

If $a = 1$, the previous curve is an Edwards curve, so twisted Edwards curves correspond to a larger set of elliptic curves.

Explicit formula for addition and doubling on twisted Edwards curves are shown in [7]. To avoid inversions, twisted Edwards curves work with projective coordinates where we consider the projective curve

$$(aX^2 + Y^2)Z^2 = Z^4 + dX^2Y^2,$$

and $(X_1 : Y_1 : Z_1)$ with $Z_1 \neq 0$ represents the affine point $(\frac{X_1}{Z_1}, \frac{Y_1}{Z_1})$.

Addition Give a pair of points $(X_1 : Y_1 : Z_1)$ and $(X_2 : Y_2 : Z_2)$, their sum $(X_3 : Y_3 : Z_3)$ can be computed as

$$\begin{aligned}
&A = Z_1 \cdot Z_2, \quad B = A^2, \quad C = X_1 \cdot X_2, \quad D = Y_1 \cdot Y_2,\\
&E = dC \cdot D, \quad F = B - E, \quad G = B + E,\\
&X_3 = A \cdot F \cdot ((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D),\\
&Y_3 = A \cdot G \cdot (D - aC),\\
&Z_3 = F \cdot G\ .
\end{aligned}$$

In summary we have:

Algorithm 3 Addition on twisted Edwards Curves

Inputs: $P = (\lambda_1 x_1, \lambda_1 y_1, \lambda_1)$ y $Q = (\lambda_2 x_2, \lambda_2 y_2, \lambda_2)$

$A = Z_1 \cdot Z_2 = \lambda_1 \cdot \lambda_2$ $\quad (1_1 1_2 \leftarrow 1_1 \cdot 1_2)$

$B = A^2 = \lambda_1^2 \cdot \lambda_2^2$ $\quad (2_1 2_2 \leftarrow 1_1 \cdot 1_2)$

$C = X_1 \cdot X_2 = \lambda_1 \lambda_2 x_1 x_2$ $\quad (1_1 1_2 \leftarrow 1_1 \cdot 1_2)$

$D = Y_1 \cdot Y_2 = \lambda_1 \lambda_2 y_1 y_2$ $\quad (1_1 1_2 \leftarrow 1_1 \cdot 1_2)$

$E = dC \cdot D = \lambda_1^2 \lambda_2^2 d x_1 x_2 y_1 y_2$ $\quad (2_1 2_2 \leftarrow 1_1 \cdot 1_2)$

$F = B - E = \lambda_1^2 \lambda_2^2 [1 - d x_1 x_2 y_1 y_2]$ $\quad (2_1 2_2 \leftarrow 2_1 - 2_2)$

$G = B + E = \lambda_1^2 \lambda_2^2 [1 + d x_1 x_2 y_1 y_2]$ $\quad (2_1 2_2 \leftarrow 2_1 + 2_2)$

$H = A \cdot F = \lambda_1^3 \lambda_2^3 [1 - d x_1 x_2 y_1 y_2]$ $\quad (3_1 3_2 \leftarrow 1_1 1_2 \cdot 2_1 2_2)$

$I = (X_1 + Y_1) \cdot (X_2 + Y_2) - C - D$
$= \lambda_1 \lambda_2 [(x_1 + y_1)(x_2 + y_2) - x_1 x_2 - y_1 y_2]$ $\quad (1_1 1_2 \leftarrow 1_1 1_2)$

$J = A \cdot G = \lambda_1^3 \lambda_2^3 (1 + d x_1 x_2 y_1 y_2)$ $\quad (3_1 3_2 \leftarrow 1_1 1_2 \cdot 2_1 2_2)$

$K = D - aC = \lambda_1 \lambda_2 (y_1 y_2 - a x_1 x_2)$ $\quad (1_1 1_2 \leftarrow 1_1 1_2 - 1_1 1_2)$

$X_3 = H \cdot I$ $\quad (4_1 4_2 \leftarrow 3_1 3_2 \cdot 1_1 1_2)$

$Y_3 = J \cdot K = \lambda_1^4 \lambda_2^4 [1 + d x_1 x_2 y_1 y_2][y_1 y_2 - a x_1 x_2]$ $\quad (4_1 4_2 \leftarrow 3_1 3_2 \cdot 1_1 1_2)$

$Z_3 = F \cdot G = \lambda_1^4 \lambda_2^4 [1 - (d x_1 x_2 y_1)^2]$ $\quad (4_1 4_2 \leftarrow 2_1 2_2 \cdot 2_1 2_2)$

Degree of $\lambda_1 \lambda_2$	Terms
$1_1 1_2$	$\{A, C, D, I, K\}$
$2_1 2_2$	$\{B, E, F, G\}$
$3_1 3_2$	$\{H, J\}$
$4_1 4_2$	$\{X_3, Y_3, Z_3\}$

Where internal collisions occur in the following cases:

$$
\begin{array}{rlrl}
x_1 x_2 = 1 & \Rightarrow A = C, & d x_1 x_2 y_1 y_2 = 1 & \Rightarrow B = E, \\
x_1 x_2 = y_1 y_2 & \Rightarrow C = D, & 2 d x_1 x_2 y_1 y_2 = 1 & \Rightarrow E = F, \\
y_1 y_2 - a x_1 x_2 = 1 & \Rightarrow A = K, & -a x_1 x_2 = 0 & \Rightarrow D = K, \\
x_1 x_2 (1 + a) = y_1 y_2 & \Rightarrow C = K, & x_1 y_2 + y_1 x_2 = 1 & \Rightarrow A = I, \\
x_1 x_2 y_1 y_2 = 0 & \Rightarrow B = F = G, J = H, & x_1 y_2 + y_1 x_2 = y_1 y_2 & \Rightarrow D = I, \\
x_1 y_2 + y_1 x_2 = y_1 y_2 - a x_1 x_2 & \Rightarrow I = K, & x_1 y_2 + y_1 x_2 = x_1 x_2 & \Rightarrow C = I, \\
y_1 y_2 = 1 & \Rightarrow A = D, Y_3 = Z_3. & &
\end{array}
$$

Doubling Given a point $(X_1 : Y_1 : Z_1)$, its doubling points is $(X_3 : Y_3 : Z_3)$ where

$$
\begin{aligned}
& B = (X_1 + Y_1)^2, \quad C = X_1^2, \quad D = Y_1^2, \quad E = aC, \\
& F = E + D, \quad H = Z_1^2 \quad J = F - 2H, \\
& X_3 = (B - C - D) \cdot J, \\
& Y_3 = F \cdot (E - D), \\
& Z_3 = F \cdot J\ .
\end{aligned}
$$

Algorithm 4 Doubling on twisted Edwards curves

Inputs: $P = (X_1, Y_1, Z_1) = (\lambda x_1, \lambda y_1, \lambda)$

$B = (X_1 + Y_1)^2 = \lambda^2 (x_1 + y_1)^2$ $(2 \leftarrow 1)$
$C = X_1^2 = \lambda^2 x_1^2$ $(2 \leftarrow 1)$
$D = Y_1^2 = \lambda^2 y_1^2$ $(2 \leftarrow 1)$
$E = aC = a\lambda^2 x_1^2$ $(2 \leftarrow 2)$
$F = E + D = \lambda^2 (y_1^2 + ax_1^2)$ $(2 \leftarrow 2 + 2)$
$H = Z_1^2 = \lambda^2$ $(2 \leftarrow 1)$
$2H = 2\lambda^2$ $(2 \leftarrow 2)$
$J = F - 2H = \lambda^2 (y_1^2 + ax_1^2 - 2)$ $(2 \leftarrow 2 - 2)$
$K = E - D = \lambda^2 (ax_1^2 - y_1^2)$ $(2 \leftarrow 2 - 2)$
$X_3 = (B - C - D) \cdot J = \lambda^4 (2x_1 y_1)(y_1^2 + ax_1^2 - 2)$ $(4 \leftarrow (2 - 2 - 2) \cdot 2)$
$Y_3 = F \cdot K = \lambda^4 (a^2 x_1^4 - y_1^4)$ $(4 \leftarrow 2 \cdot 2)$
$Z_3 = F \cdot J = \lambda^4 (y_1^2 + ax_1^2)(y_1^2 + ax_1^2 - 2)$ $(4 \leftarrow 2 \cdot 2)$

In summary we have:

Degree of λ	Terms
2	$\{B, C, D, E, F, H, 2H, J, K\}$
4	$\{X_3, Y_3, Z_3\}$

The internal collisions occur for the following cases:

$$
\begin{aligned}
y_1 &= 0 && \Rightarrow B = C, E = F, & x_1 &= \pm 1 && \Rightarrow C = H \\
2x_1 + y_1 &= 0 && \Rightarrow B = C, & x_1^2 - y_1^2 &= ax_1^2 - 2 && \Rightarrow C = J \\
x_1 &= 0 && \Rightarrow B = D, B = F, & y_1^2 &= ax_1^2 && \Rightarrow D = E \\
x_1 + 2y_1 &= 0 && \Rightarrow B = D, & ax_1^2 &= 0 && \Rightarrow D = F \\
(x_1 + y_1)^2 &= ax_1^2 && \Rightarrow B = E, & y_1^2 &= 1 && \Rightarrow D = H \\
x_1 + 2y_1 - ax_1 &= 0 && \Rightarrow B = F, & ax_1^2 - 2 &= 0 && \Rightarrow D = J \\
x_1 + y_1 &= 1 && \Rightarrow B = H, & a = 1 \wedge x_1 &= \pm 1 && \Rightarrow E = H \\
x_1 + y_1 &= -1 && \Rightarrow B = H, & y_1^2 - 2 &= 0 && \Rightarrow E = J \\
x_1 &= \pm y_1 && \Rightarrow C = D & y_1^2 + ax_1^2 &= 1 && \Rightarrow F = H \\
a &= 1 && \Rightarrow C = D, & y_1^2 + ax_1^2 &= 3 && \Rightarrow H = J \\
x_1^2 &= y_1^2 && \Rightarrow C = D, & 2x_1 y_1 &= y_1^2 + ax_1^2 && \Rightarrow X_3 = Z_3 \\
x_1^2 &= ax_1^2 && \Rightarrow C = E, & y_1 &= \pm 1 && \Rightarrow Y_3 = Z_3 \\
y_1^2 + ax_1^2 &= x_1^2 && \Rightarrow C = F, & & &&
\end{aligned}
$$

$$
\begin{gathered}
x_1(x_1 + 2y_1 - ax_1) = 2 \Rightarrow B = J, \\
(ax_1^2)^2 - (y_1^2)^2 = 2x_1 y_1 (y_1^2 + ax_1^2 - 2) \Rightarrow X_3 = Y_3
\end{gathered}
$$

4 SVA on New Edwards Curves

Recently, several research groups have proposed specific curves in Edwards or Twisted Edwards form for use cryptographic application (and possibly future standards) [3, 5, 8, 9, 22]. Our analysis indicates that all these curves have points of SVA type.

In the following table we present the SVA points found

Table 5: SVA Points for Doubling in Edwards curve, birationally equivalent to the Montgomery Curves

E: $y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$					
Curve	$y = 2x$	$(x+y)^2 = 1$	$y^2 = 2$	$xy = 1$	$x^2 = y^2$
M-159	✓	✓	∅	✓	✓
M-191	✓	✓	∅	✓	✓
M-221	∅	✓	∅	∅	✓
Curve255519	✓	✓	∅	∅	✓
M-383	∅	∅[13]	∅	✓	✓
M-511	∅	∅[14]	∅	∅	✓

Table 6: SVA Points for Doubling on Edwards Curves

$E : x^2 + y^2 = 1 + dx^2y^2$ over $\mathbb{F}_p$					
Curve	$y = 2x$	$(x+y)^2 = 1$	$y^2 = 2$	$xy = 1$	$x^2 = y^2$
E-157	∅	✓	∅	✓	∅
E-168	✓	∅	✓	∅	∅
E-191	∅	✓	∅	∅	∅
E-222	∅	✓	∅	∅	∅
Curve1174	∅	✓	∅	✓	∅
E-382	✓	∅	✓	✓	∅
Curve41417	✓	∅	∅	∅	✓
E-448-Goldilocks	✓	∅	∅	∅	∅
E-521	∅	∅	∅	✓	∅

5 Conclusion

In this paper we analyzed the existence of SVA points in Edwards model of elliptic curves. Our analysis indicates that all curves that have been proposed for cryptographic application are vulnerable to these attacks.

[13] Point of order two.
[14] Point of order two.
[15] For details see [29].

Table 7: SVA Points for Doubling in Twisted Edwards Curves

$\mathcal{E}_d : -x^2 + y^2 = 1 + dx^2y^2$ over $\mathbb{F}_p$					
Curves $\mathbb{F}_p$	ZVA[15]	SVA			
	$x = y$, $x = -y$	$x = -y/2$	$x = -2y$	$x = 1$	$(x + y)^2 = 1$
ed-256-mont	∅	∅	∅	∅	✓
ed-254-mont	∅	∅	∅	∅	✓
ed-256-mers	∅	∅	∅	∅	✓
ed-255-mers	∅	✓	∅	∅	∅
ed-384-mont	∅	∅	✓	✓	∅
ed-382-mont	∅	∅	∅	✓	✓
ed-384-mers	∅	∅	∅	∅	✓
ed-383-mers	∅	∅	∅	✓	✓
ed-512-mont	∅	∅	∅	✓	✓
ed-510-mont	∅	✓	∅	∅	✓
ed-512-mers	∅	∅	✓	✓	✓
ed-511-mers	∅	∅	∅	✓	∅
ed-521-mers	∅	∅	✓	∅	✓
Ted37919	∅	∅	∅	✓	∅

References

[1] T. Akishita and T. Takagi, *Zero-Value Point Attacks on Elliptic Curve Cryptosystem,* Information Security, ISC 2003, LNCS 2851, pp. 218–233, Springer 2003.

[2] T. Akishita and T. Takagi, *On the Optimal Parameter Choice for Elliptic Curve Cryptosystems using Isogeny.* PKC 2004, LNCS 2947, pp. 346–359, 2004.

[3] D. Aranha, P. Barreto, G. Pereira and J. Ricardini, *A Note on High-Security General-Purpose Elliptic Curves.* IARC Cryptology ePrint Archive, Report 2013/647, 2013. http://eprint.iacr.org/

[4] R. Avanzi, *Side Channel Attacks on Implementations of Curve-Based Cryptographic Primites.* Cryptology ePrint Archive Report 2005/017, available from http://eprint.iacr.org/

[5] D. J. Bernstein, *Curve25519: New Diffie-Hellman Speed Records.* Public Key Cryptography – PKC 2006, volume 3958 of Lecture Notes in Computer Science, pp. 207–228, 2006,

[6] D. J. Bernstein and T. Lange, *Faster Addition and Doubling on Elliptic Curves.* ASIACRYPT 2007, LNCS 4833, pp. 29–50, Springer 2007.

[7] D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, *Twisted Edwards Curves.* AFRICACRYPT 2008, LNCS 5023, pp. 389–405, Springer 2008.

[8] D. J. Bernstein, M. Hamburg, A. Krasnova, and T. Lange, *Elligator: Elliptic-Curve Points Indistinguishable from Uniform Random*

Strings. IACR Cryptology ePrint Archive, report 2013/325, 2013. http://eprint.iacr.org/

[9] J. W. Bos, C. Costello, P. Longa, and M. Naehrig, *Selecting Elliptic Curves for Cryptography: An Efficiency and Security Analysis.* IACR Cryptology ePrint Archive, report 2014/130, 2014. http://eprint.iacr.org/

[10] S. Chari, J. R. Rao, and P. Rohati, *Template attacks.* Cryptographic Hardware and Embedded Systems – CHES 2003 LNCS 2523 pp.13-28 Springer, 2003.

[11] B. Chevallier-Mames. *Self-Randomized Exponentiation Algorithms.* CT-RSA 2004, LNCS 2964, pp. 236-249, Springer-Verlag 2004.

[12] M. Ciet and M. Joye, *(Virtually) Free Randomization Techniques for Elliptic Curve Cryptography.* ICICS 2003, LNCS 2836, pp. 348-359, Springer-Verlag 2003

[13] C. Clavier, B. Feix, G. Gagnerot, M. Roussellet, and V. Verneuil, *Improved Collision-Correlation Power Analysis on First Order Protected AES.* CHES 2011, LNCS 6917, pp. 49–62, Springer 2011.

[14] C. Clavier and M. Joye, *Universal Exponentation Algorithm.* Cryptographic Hardware and Embedded Systems – CHES 2001, LNCS 2162, pp. 300-308, Springer-Verlag, 2001.

[15] J. Coron, *Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems.* Cryptographic Hardware and Embedded Systems – CHES 1999, LNCS 1717, pp. 392–302, Springer, 1999.

[16] N. M. Ebeid, *Key Randomization Countermeasures to Power Analysis Attacks on Elliptic Curve Cryptosystems.* Phd.D. Electrical and Computer Engineering, University of Waterloo.

[17] H. M. Edwards, *A Normal Form for Elliptic Curves.* Bull. Am. Math. Soc., New Ser., **44**(3), pp. 393–422, 2007.

[18] B. Feix and V. Verneuil. *There's Something about m-ary, Fixed-Point Scalar Multiplication Protected against Physical Attacks.* INDOCRYPT 2013, LNCS 8250, pp. 197–214, Springer 2013.

[19] K. Gandolfi, C. Mourtel, and F. Olivier, *Electronic Analysis: Concrete Results.* Cryptographic Hardware and Embedded Systems – CHES 2001, LNCS 2162, pp. 251–261, Springer, 2001.

[20] D. Giry and J.-J. Quinsquater, *Bluekrypt Cryptographic Key Length.* Recommendation 2011, v26.0, April 18, http://www.keylength.com/.

[21] L. Goubin, *A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems.* Public Key Cryptography, LNCS 2567, pp. 199–210, Springer, 2003.

[22] M. Hamburg, *Ed448-Goldilocks, Fast, Strong Elliptic Curve Cryptography.* http://ed448goldilocks.sourceforge.net/ .

[23] M. Joye, *Elliptic Curves and Side-channel Analysis.* ST Journal of System Research, 4(1), pp. 283-306, 2003.

[24] M. Joye and C. Tymen, *Protections against Differential Analisis for Elliptic Curve Cryptography.* Cryptographic Hardware and Embedded Systems – CHES 2001, LNCS 2162, pp. 377-390. Springer-Verlag, 2001.

[25] N. Koblitz, *Elliptic Curve Cryptosystems.* Mathematics of Computation, **48**, pp. 203–209, 1987.

[26] P. Kocher, *Timing Attacks on Implementation of Diffie-Hellman RSA, DSS and other Systems.* Advances in Cryptology – CRYPTO 1996, LNCS 1109, pp. 104–113, Springer, 1996.

[27] P. Kocher, J. Jaffe, and B. Jun, *Differential power Analisis.* Advances in Cryptology –CRYPTO 1999, LNCS 1666, pp. 388–397, Springer, 1999.

[28] P. Longa, *Post-Snowden Elliptic Curve Cryptography.* Microsoft Research http://research.microsoft.com/en-us/people/plonga/

[29] S. Martínes, D. Sadornil, J. Tena, R. Tomàs, and M. Valls. *On Edwards Curves and ZVP-Attacks* Applicable Algebra in Engineering Communication and Computing, Vol 24, pp. 507–517, Springer-Verlag 2013.

[30] V. S. Miller, *Use of Elliptic Curves in Cryptography.* Advances in Cryptology – CRYPTO 1985. LNCS 218, pp. 417-426, Springer, 1986.

[31] J. Miret, D. Sadornil, J. Tena, R. Tomàs, and M. Valls. *Isogeny Cordillera Algorithm to obtain Cryptographically good Elliptic Curves.* Australasian Information Security Workshop: Privacy Enhancing Tecnologies (AISW), CRPIT vol. 68, pp. 127-131, 2007.

[32] C. Murdica, S. Guilley, J-L. Danger, P. Hoogvourst, and D. Naccache, *Same Value Power Analysis Using Special Point on Elliptic Curves.* COSADE 2012, LNCS 7275, pp. 183-198, Springer-Verlag 2012.

[33] D. Naccache, N. P. Smart, and J. Stern. *Projective Coordinates Leak.* EUROCRYPT 2004, LNCS 3027, pp. 257–267. 2004.

[34] J-J. Quisquater, and D. Samyde, *Electromagnetic analysis (EMA): Measures and Coutermeasures for Smard Cards.* Smart Card Programming and Security – E-SMART 2001, LNCS 2140, pp. 200–210, Springer, 2001.

[35] K. Schramm, T. Wollinger, and C. Paar, *A New Class of Collision Attacks and Its Application to DES.* FSE 2003. LNCS, vol 2887, pp. 206-222 Springer 2003.

[36] STANDARDS FOR EFFICIENTE CRYPTOGRAPHY, *SEC 2: Recommended Elliptic Curve Domain Parameters.* Certicom Corp. Version 2.0, January 2010.

[37] N. Smart. *An Analysis of Goubin's Refined Power Analysis Attack.* CHES 2003, LNCS 2779, pp. 281-290, 2003.

[38] N. P Smart, E. Oswald, and D. Page, *Randomised Representations.* IET Inf. Secur., 2008, Vol. 2, No 2, pp. 19-27, 2008.

[39] E. G. Strauss. *Addition chains of vectors (problem 5125).* American Mathematical Monthly, vol. 70, pp. 806–808, 1964.

[40] E. Trichina and A. Belleza, *Implementation of Elliptic Curve Cryptography with Built-In Counter Measures against Side Channel Attacks.* Cryptographic Hardware and Embedded Systems, CHES –2002, LNCS 2523, pp. 98–113, Springer 2002.

[41] M. Tunstall and M. Joye, *Coordinate Blinding over Large Prime Fields.* CHES-2010, LNCS 6225, pp. 443-445, Springer 2010.