# An Analysis of ZVP-Attack on ECC Cryptosystems

Claude Crépeau *, Raza Ali Kazmi *

School of Computer Science, McGill University,
Montréal, QC, Canada. {crepeau,rkazmi}@cs.mcgill.ca

**Abstract** Elliptic curve cryptography (ECC) is an efficient public cryptosystem with a short key size. For this reason it is suitable for implementing on memory-constraint devices such as smart cards, mobile devices, etc. However, these devices leak information about their private key through side channels (power consumption, electromagnetic radiation, timing etc) during cryptographic processing. In this paper we have examined countermeasures against a specific class of side channel attacks (power analysis) called Zero-Value Point Attack (**ZVP**), using elliptic curve isomorphism and isogeny. We found that these methods are an efficient way of securing cryptographic devices using ECC against **ZVP** attack. Our main contribution is to extend the work of Akishita and Takagi [3,2] to binary fields. We also provide a more detail analysis of the **ZVP** attack over prime fields.

**Keywords.** Elliptic Curve Cryptography, Side Channel Attacks, Zero-Value Point Attack, Isomorphism, Isogeny.

## 1 Introduction

The advantage of elliptic curve based cryptosystems over other cryptosystems (**RSA**, etc.) is their short key size. For this purpose it is suitable for implementing on memory-constrained devices such as smart cards, mobile devices, etc. A cryptographic device uses a private key to process input information. The designer of the cryptosystem assumes that the attacker has pairs of plaintext/ciphertext, key sizes, but other secrets will be manipulated in closed and safe computing environments. However these devices leak information about the private key through side channels (power consumption, electromagnetic radiation, timing, etc.) during cryptographic processing. The term "side channel" is used to describe the leakage of unintended information during cryptographic processing from a supposedly tamper-resistant device, such as smart cards. Hence, the side channel attacks are practical attacks as opposed to theoretical attacks (e.g. differential cryptanalysis attack on **DES**, etc). There are many kinds of side channel attacks such as timing attacks, power analysis attacks, electromagnetic attacks, etc. The side channel attack we are interested in is power analysis. More specifically we are interested in Zero-Value Point Attack (**ZVP**), which is a kind of power analysis attack. Note that we will assume throughout this chapter that we are working on elliptic curves defined over prime fields $\mathbf{F}_p$, with $p > 3$ or binary fields $\mathbf{F}_{2^m}$, $m \geq 1$. Akishita and Takagi [3] and their predecessors, Coron [6], Goubin [8], Okeya and Sakurai [14] proposed power analysis attacks on elliptic curve cryptosystems that would allow an adversary to recover the private key by monitoring the power consumption of cryptographic devices.

In this paper we assess the application of elliptic curve isomorphisms and isogenies (rational homomorphisms between elliptic curves) for defence against the **ZVP** attack [3].

* Supported in part by Québec's FQRNT, Canada's NSERC, CIFAR, and QuantumWorks.

## 2 Contribution of the Paper

In [2], Akishita and Takagi examined the isogeny defence against **ZVP** attack on some standard curves over prime fields (they examined **SECG** standards but not **NIST**). They also did not discussed any defence against **ZVP** attack or existence of **ZVP** points over $\mathbf{F}_{2^m}$ [2,3]. Over prime fields they only examined ZVP points $3x^2 + a = 0$ and $(0, y)$ and did not consider the point $5x^4 + 2ax^2 - 4bx + a^2 = 0$. The contributions of this paper are the following:

- We found that most standard curves (**NIST**,**SECG**) over $\mathbf{F}_{2^m}$ have **ZVP** points. We showed that a certain class of standard curves over $\mathbf{F}_{2^m}$ can be defended efficiently against **ZVP** attack using elliptic curve isomorphisms. We show that the remaining classes of curves can be defended efficiently using isogenies. We also investigate the isogeny defence against **ZVP** attack for both **SECG** and **NIST**) curves and found that if we use an isogeny defence against all **ZVP** points, then in some curves the isogeny degree increases dramatically.
- We calculate the additional cost of the isogeny and an elliptic curve isomorphism defence for standard curves.[1]

## 3 Background material

An elliptic curve $E(\mathbf{K})$(in Weierstrass form) over a field **K** is a set of solutions of an equation of the form $E(\mathbf{K}) : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where $a_i \in \mathbf{K}$. If $char(\mathbf{K}) \neq 2, 3$ the equation can be transformed to $E(\mathbf{K}) : y^2 = x^3 + a_1x + b_1$, $a_1, b_1 \in \mathbf{K}$. If $char(\mathbf{K}) = 2$ equation can be transformed to $E(\mathbf{K}) : y^2 + xy = x^3 + a_2x^2 + b_2$, $a_2, b_2 \in \mathbf{K}$. The set of points on $E(\mathbf{K})$ form an additive abelian group with $P_\infty$ (the point at infinity) as the identity element (for more details see [4,5,16]).

### Projective Coordinates

In order to avoid costly inversions, elliptic curves point doubling (ECDBL) and point addition (ECADD) is done using projective coordinates. For more details see appendices A and B.

### Scalar Multiplication

The operation of adding a point $P$ to itself $d$ times is called scalar multiplication by $d$ and denoted by $[d]P$. The simplest and oldest efficient method for scalar multiplication is called **binary method**.

**Algorithm 1.** (Binary Method)

- **Input** $P$, $d = \sum_{j=0}^{l-1} d_j 2^j$, where $d_j \in \{0, 1\}$.
- $Q \leftarrow P$.
  1. for $i$ from $l - 2$ to 0 do
     (a) $Q \leftarrow \mathrm{ECDBL}(Q)$.
        i. if $d_i = 1$ then $Q \leftarrow \mathrm{ECADD}(Q, P)$.
     (b) **Output** $Q$.

Let $\mathbf{A}$ and $\mathbf{D}$ denote the cost of point doubling (ECDBL) and point addition (ECADD) in finite fields. The algorithm 1 requires in total $[(l-1)]\mathbf{D} + [(W-1)]\mathbf{A}$ field operations, where $W$ is the weight (number of 1's) in the binary representation of $d$.

[1] In this paper computations are carried out using Magma computer algebra system.

## Elliptic Curve Isomorphism over $\mathbf{F}_{2^m}$

Let $E_1 : y_1^2 + x_1 y_1 = x_1^3 + A_1 x_1^2 + B_1$ and $E_2 : y_2^2 + x_2 y_2 = x_2^3 + A_2 x_2^2 + B_2$ be two elliptic curves over $\mathbf{F}_{2^m}$ with $B_i \neq 0$ for $i = 1, 2$. Then $E_1$ and $E_2$ are isomorphic over $\mathbf{F}_{2^m}$ if and only if there exists $s \in \mathbf{F}_{2^m}$ such that $A_2 = A_1 + s + s^2$ and $B_2 = B_1$. The *j-invariants* are given by $j_i = B_i^{-1}$, for $i = 1, 2$. Furthermore, isomorphisms $\varphi$ and $\varphi^{-1}$ are given by [13]

$$\varphi : E_1 \longrightarrow E_2, \;\; \varphi\,(x, y) \longmapsto (x, y + sx);$$

$$\varphi^{-1} : E_2 \longrightarrow E_1, \;\; \varphi^{-1}\,(x', y\prime) \longmapsto (x', y' + sx').$$

## Isogeny

Let $\overline{\mathbf{K}}$ be the algebraic closure of a finite field $\mathbf{K}$. The field of rational functions in two variables over $\overline{\mathbf{K}}$ is

$$\overline{\mathbf{K}}(x, y) = \left\{ \frac{f(x,y)}{g(x,y)} \middle| f(x,y), g(x,y) \in \overline{\mathbf{K}}[x, y] \textit{ and } g(x,y) \neq 0 \right\} .$$

Let $E_1(\overline{\mathbf{K}})$ and $E_2(\overline{\mathbf{K}})$ be two elliptic curves. An **isogeny** $I$ is a homomorphism that is given by rational functions

$$I(x, y) : E_1(\overline{\mathbf{K}}) \longrightarrow E_2(\overline{\mathbf{K}})$$

where $I(x, y) = (R_1(x, y), R_2(x, y))$ and $R_i(x, y) \in \overline{\mathbf{K}}(x, y)$ for *i= 1, 2*. Two elliptic curves are called isogenous if there exists an isogeny between them. The degree of $I(x, y)$ denoted $deg(I)$ is the number of elements in $ker(I) = \{(x, y) \in E_1(\overline{\mathbf{K}}) | I(x, y) = P_\infty\}$. The $ker(I)$ is a finite set. An isogeny is fully determined by its kernel. Over a finite field if two elliptic curves are isogenous, then they have equal number of points.To every non-constant isogeny $I : E_1 \longrightarrow E_2$ of degree $l$, there exist a unique dual isogeny $\hat{I} : E_2 \longrightarrow E_1$ of degree $l$ such that for all points $P_1 \in E_1$ and $P_2 \in E_2$.

$$\hat{I}(I(P_1)) = P_1 \quad and \quad I(\hat{I}(P_2)) = P_2$$

For details see [4,1].

# 4 Elliptic Curve Cryptosystems

This scheme is analogous to El-Gamal encryption [7].

- **Common Parameters**
  - An elliptic curve *E* over $\mathbf{F}_p$ or $\mathbf{F}_{2^m}$.
  - The order of $\#E$ must be divisible by a large prime $q$.
  - $P \in E$.
- **Private Key**
  - $d \in [1, q-1]$ chosen randomly.
- **Public Key**
  - $Q = [d]P$.

– **Encryption Of message** $m$
  - Pick a random $n \in [1, q-1]$.
  - Compute the points $(x_1, y_1) = [n]P$ and $(x_2, y_2) = [n]Q$.
  - Compute $c = x_2 + m$.
  - Output ciphertext $(x_1, y_1, c)$.
– **Decryption**
  - Receive ciphertext $(x_1, y_1, c)$.
  - Compute $(x, y) = [d](x_1, y_1)$ and $m = c - x$.

# 5 Previous Work

## 5.1 Power Analysis Attack

In a power analysis attack the side channel is the device's power consumption. A power analysis attack works by exploiting the fact that a tamper-resistant device such as a smart card consumes different amount of power if it is processing 0 or 1 [10]. There are two types of power analysis attacks: one is called **Simple Power Analysis** (**SPA**) and the other is **Differential Power Attack** (**DPA**).

## 5.2 Simple Power Analysis

A simple power analysis attack consists of observing the power consumption of one single execution of a cryptographic algorithm. We assume that the scalar multiplication is computed by Algorithm 1. Let $E$ be an elliptic curve and $P$ be a point on it. Suppose an attacker knows $P$, then by monitoring the power consumption during the computation of $Q = [d]P$, he/she can recover the private key $d$, since we perform step 3 only if $d_i = 1$, the power consumption will be more when $d_i = 1$ thus revealing the bits of the private key $d$. Algorithm 1 can easily be modified so that step 3 is performed no matter what the secret bit is.

**Algorithm 2.** (Always-Add-Double Binary Method)

– **Input** $P$, $d = \sum_{j=0}^{l-1} d_j 2^j$, where $d_i \in \{0, 1\}$.
– $Q[0] \leftarrow P$.
  1. for $i$ from $l-2$ to 0 do
     (a) $Q[0] \leftarrow \mathrm{ECDBL}(Q[0])$.
     (b) $Q[1] \leftarrow \mathrm{ECADD}(Q[0], P)$.
     (c) $Q[0] \leftarrow Q[d_j]$.
  2. **Output** $Q[0]$.

The computational cost of this countermeasure is $(l-1)\mathbf{A} + (l-1)\mathbf{D}$. The algorithm 2 is secure against the **SPA** attack defined above. We assume that algorithm 2 is performed in constant time (i.e. The time for each *i-th* loop is the same). Otherwise the implementation can be subject to timing attacks [11].

### 5.3 Differential Power Analysis Attack (DPA) and Countermeasures

A **DPA** is a more powerful attack [6]. It consists of performing statistical analysis of several execution of the same algorithm with possibly many different inputs. Even if the method is secure against **SPA** it might not be secure against the **DPA**. The **DPA** attacker gathers many power consumptions $[d]P_i$, $i = 1, 2, ...$ and detects the spike arises from the correlation function based on the specific bit of $[d]P_i$. The algorithm 2 is insecure against **DPA** attack, because the sequence of points generated by it is deterministic and the attacker can find the correlation for a specific bit. In [6] Coron proposed three countermeasures against the **DPA** attack[6]: 1) Randomization of the private key $d$. 2) Adding a random point $R$ to the base point $P$. 3) Using randomized projective coordinates. In addition, Joye and Tymen [9] also proposed two countermeasures against Coron's **DPA**. 4) In the first countermeasure, they chose a random isomorphic elliptic curve and computed the scalar multiplication over that curve. 5) In the other countermeasure they computed the scalar multiplication in an isomorphic field which is also chosen randomly. In [14], Okeya and Sakurai showed that Coron's first and second countermeasures are not secure against the **DPA** attack, as they do not properly randomize the private key $d$ or base point $P$.[2] The last three countermeasures (randomized projective coordinates, random elliptic curve isomorphism and random field isomorphism) seem to be effective against **DPA** attack. Among this the Coron's third countermeasure is the most efficient one [1]. Due to space limitation we will only describe Coron's third countermeasure. For all other see [6,14].

## Third countermeasure: Randomization in Projective Coordinates

Let $P = (x, y)$, be the base point. The computation of $Q = [d]P$ is done as follows.

1. Map affine coordinates $(x, y)$ to projective coordinates $X \leftarrow x, Y \leftarrow y, Z \leftarrow 1$.
2. Choose a random $\alpha \in \mathbf{K}^*$. If $\mathbf{K}$ is a binary field, then set $P' = (\alpha x, \alpha^2 y, \alpha)$ and if $\mathbf{K}$ is a prime field, then set $P' = (\alpha^2 x, \alpha^3 y, \alpha)$.
3. Compute $Q' = [d]P' = (X, Y, Z)$ using algorithm 2.
4. Compute $Q$ in affine coordinates by setting $x \longleftarrow X/Z$ , $y \longleftarrow X/Z^2$ for binary fields or $x \longleftarrow X/Z^2$ , $y \longleftarrow X/Z^3$ for prime fields.
5. **Output** $Q = (x, y)$.

Let $M$ and $R$ denote the computational cost of multiplication and random number generation in finite fields. The computational cost of this countermeasure is $(l-1)]\mathbf{A} + [(l-1)]\mathbf{D} + 3\mathbf{M} + \mathbf{R}$. This countermeasure makes the **DPA** attack infeasible since the representation of point $P$ in projective coordinates remains unknown to the attacker. However, Goubin observed that points $(0, y)$ and $(x, 0)$(called special points) cannot be properly randomized by any of these anti-**DPA** methods(randomized projective coordinates, random elliptic curve isomorphism and random field isomorphism) [8]. Hence if these special points lie on the elliptic curve then they can be used to launch a **DPA** attack (called refined power analysis attack) on the elliptic curve scalar multiplication [8]. In standards, $(\#E = h \cdot q)$, where $q$ is a large prime and $h$

[2] These attacks could be thwarted by modifying these countermeasures. However, this will increase the cost of countermeasure 1 from $[(l-1)+19]\mathbf{A} + [(l-1)+19]\mathbf{D} + \mathbf{R}$ to $[2(l-1)]\mathbf{A} + [2(l-1)]\mathbf{D} + \mathbf{R}$ and countermeasure 2 from $[(l+1)]\mathbf{A} + [(l+1)]\mathbf{D} + \mathbf{R}$ to $[(l+2n-1)]\mathbf{A} + [(l+2n-1)]\mathbf{D} + \mathbf{R}$, where $n$ suggested to be $20$ [14].

is a small integer called cofactor. Nigel Smart pointed out that special points of small order can easily be dealt by first computing $Q \leftarrow [h]P$ and if $Q \neq P_\infty$, then compute $[d]P$ using algorithm 2. Hence, this way no point of small order will ever enter the scalar multiplication algorithm with private key $d$. Note that over binary fields $(0, y)$ has order 2 and over prime fields $(x, 0)$ has order 2. For points of large order Smart proposed a defence using isogenies [15].

## 6 Zero-Value Point Attack

In [3], Toru Akishita and Tsuyoshi Takagi proposed an attack called **Zero Value Point Attack**, which is a generalization of Goubin's refined power analysis attack [8]. They show that even if elliptic curves (over prime fields) have no *special points* $(x, 0)$ and $(0, y)$, they can still have points called *zero value points* (**ZVP**), for which auxiliary registers takes zero value and these points cannot be randomized by the third countermeasure (randomized projective coordinates). If the **ZVP** points lie on the elliptic curves, then they can be used to launch a **DPA** attack on elliptic curve scalar multiplication. However, unlike Goubin's attack, the **ZVP** attack depends strongly on the implementation of a scalar multiplication algorithm [3]. We assumed that scalar multiplication is computed by Algorithm 2 and **ECDBL** and **ECADD** are implemented as described in appendices A and B. Note that we will only discuss defence against **ZVP** points from **ECDBL** since finding **ZVP** point from **ECADD** is believed to be a hard problem for both binary and prime fields [1,3].

### 6.1 Outline of the Attack

The task of **ZVP** attack is to learn the private key $d$ by adaptively choosing the base point $Q$. The attacker breaks the private key from the most significant bit. The second most significant bit $d_{l-2}$ can be broken by checking whether one of formulae **ECDBL**$(2Q)$, **ECADD**$(2Q, Q)$, **ECDBL**$(3Q)$ and **ECADD**$(3Q, Q)$ is computed. If we can generate the zero-value register for these addition formulae, we can recover the $d_{l-2}$ bit of the key. If **ECDBL**$(2Q)$ or **ECADD**$(2Q, Q)$ has zero-value register, then $d_{l-2} = 0$ and if **ECDBL**$(3Q)$ or **ECADD**$(3Q, Q)$ is zero then $d_{l-2} = 1$. Now assume that $(d_{l-1}, \ldots, d_{l-i+1})$ bits of $d$ are known to attacker. He can recover the $d_{l-i}$ bit by checking whether one of **ECDBL**$(2kQ)$, **ECADD**$(2kQ, Q)$, **ECDBL**$((3k+1)Q)$ and **ECADD-F**$((3k+1)Q, Q)$, where $d = \sum_{j=i+1}^{l-1} d_j 2^{j-i-1}$. We know that $d_{l-i} = 0$ if **ECDBL**$(2kQ)$, **ECADD-F**$(2kQ, Q)$ has a zero-value register and $d_{l-i} = 1$ if **ECDBL**$((3k+1)Q)$ or **ECADD**$((3k+1)Q, Q)$, has a zero-value register. Therefore if we find a point $P$ that takes the zero-value register at **ECDBL**, we can use the base point $Q = (c^{-1} \mod \#E)P$ for some integer c for this attack. On the other hand, in order to use the zero-value register at **ECADD**$(2Q, Q)$, the base point $Q$ that causes the zero-value register at **ECADD**$(cQ, Q)$ must be found. Thus the attacker has to find a point $Q$ which cause the zero-value register at **ECDBL**$(cQ)$ or **ECADD**$(cQ, Q)$ for some integer $c$.

### 6.2 Zero-Value Point Value Attack over Binary Fields $\mathbf{F}_{2^m}$

**Theorem 1.** Let $E : y^2 + xy = x^3 + Ax^2 + B$ be an elliptic curve over $\mathbf{F}_{2^m}$ such that $B \neq 0$. The elliptic curve $E$ has a zero value point $P = (x, y)$ from **ECDBL** if and only if one of the following conditions are satisfied: $(1)\, x^2 + y = 0,\ (2)\, Ax^2 + y^2 = 0,\ (3)\, y^2 + B = 0,\ (4)\, y(P) =$

$0$ *or* $y([2]P) = 0, (5)\, x(P) = 0$ *or* $x([2]P) = 0$.[3] Moreover, the zero-value points are not randomized by Coron's third countermeasure (randomized projective coordinates).

**Proof**. Let $P_1 = (X_1, Y_1, Z_1) \neq P_\infty$ be the corresponding point in projective coordinates. Let $P_2 = \mathbf{ECDBL}(P_1)$. The **ECDBL** (see appendix) has a zero value register if and only if one of the following values are zero: $X_1$, $Y_1$, $Y_1$, $X_2$, $Y_2$, $Z_2$, $AZ_2 + Y_1^2$, $Y_1^2 + BZ_1^4$, $AZ_2 + BZ_1^4$, $AZ_2 + Y_1^2 + BZ_1^4$. $AZ_2 + Y_1^2 + BZ_1^4 = 0 \iff AX_1^2Z_1^2 + Y_1^2 + BZ_1^4 = 0$, which is in affine coordinate is $Ax^2Z_1^4 + y^2Z_1^4 + BZ_1^4 = 0$.

| Name of Curve | $x^2 + y = 0$ | $Ax^2 + y^2 = 0$ | $y^2 + B = 0$ | $(x, 0)$ | Curve Order |
|---|---|---|---|---|---|
| sect113r1 | no | no | yes(large) | no | $2 \times Prime$ |
| sect113r2 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| sect131r1 | yes(large) | no | yes(large) | yes(large) | $2 \times Prime$ |
| sect131r2 | no | no | no | yes(large) | $2 \times Prime$ |
| sect163k1 | no | no | no | no | $4 \times Prime$ |
| sect163r1 | yes(large) | no | yes(large) | no | $2 \times Prime$ |
| sect163r2 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| sect193r1 | yes(large) | no | no | yes(large) | $2 \times Prime$ |
| sect193r2 | yes(large) | yes(large) | no | no | $2 \times Prime$ |
| sect233k1 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| sect233r1 | no | no | no | no | $2 \times Prime$ |
| sect239k1 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| sect283k1 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| sect283r1 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| sect409k1 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| sect409r1 | no | no | no | no | $2 \times Prime$ |
| sect571k1 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| sect571r1 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| B-163 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| K-163 | no | no | no | no | $2 \times Prime$ |
| B-233 | no | no | no | no | $2 \times Prime$ |
| K-233 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| B-283 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| K-283 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| B-409 | no | no | no | no | $2 \times Prime$ |
| K-409 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |
| B-571 | no | yes(large) | no | yes(large) | $2 \times Prime$ |
| K-571 | yes(small) | yes(small) | yes(small) | yes(small) | $4 \times Prime$ |

**Table 1. SECG** and **NIST** curves over $\mathbf{F}_{2^m}$ and **ZVP** points from **ECDBL-F**$_{2^m}$

This implies $Ax^2 + y^2 + B = 0 = Ax^2 + y^2 + (y^2 + xy + x^3 + Ax^2) = 0 = 2Ax^2 + 2y^2 + xy + x^3 \equiv xy + x^3 \mod 2$. Hence, $(x^2 + y) = 0$ or $x = 0$. Which is condition

[3] For a point $P = (x, y)$ we denote its $x$-coordinate by $x(P)$ and $y$-coordinate by $y(P)$

(1) and $X_1 = 0 \implies x(P) = 0$ and $X_2 = 0 \implies x(2P) = 0$ which is condition $(4)$ and $Y_1 = 0 \implies x(P) = 0$ and $Y_2 = 0 \implies x(2P) = 0$ which is condition $(5)$. $AZ_2 + Y_1^2 = 0 \iff AX_1^2Z_1^2 + Y_1^2 \implies Ax^2 + y^2 = 0$ in affine coordinate which is condition $(2)$. Similarly $Y_1^2 + BZ_1^4 \implies y^2 + B = 0$ gives is condition $(3)$.

Preventing **ZVP** point attack or existence of **ZVP** points on standard curves (**SECG, NIST**) over $\mathbf{F}_{2^m}$ was not discussed in [2,3]. We have found that $15$ out of $18$ **SECG** curves and $7$ out of $10$ **NIST** have **ZVP** points $(x, y)$ from **ECDBL** which satisfy at least one of the conditions of theorem 1 (see Table 1). Note that each row of table 1 tell us whether or not the elliptic curve from standards has any **ZVP** point on it. In parenthesis ( ) we list if the **ZVP** point is of large or small order. As discuss at the end of section 5 that **ZVP** points of small order can easily be protected. Therefore, we do not consider **ZVP** points of form $(0, y)$ (they will always have order 2) in table above. We notice that Koblitz curves[4] (in **SECG** names ending with k1 *e.g.* sectk1, and in **NIST** *starting with* K e.g. K-571) have **ZVP** points of small order only.

### 6.3 Finding ZVP Points from ECDBL over Binary Fields $\mathbf{F}_{2^m}$

In this section we will discuss how to find **ZVP** from **ECDBL** over $\mathbf{F}_{2^m}$. Let $E : y^2 + xy = x^3 + Ax^2 + B$ be an elliptic curve over $\mathbf{F}_{2^m}$. Condition(1) $x^2 + y = 0$, let $P \in E$ be such that $x^2 = y \implies x^4 + Ax^2 + B = 0$. The solution of this polynomial can easily be found efficiently [5]. Condition(2) $Ax^2 + y^2 = 0$, let $P \in E$ such that $y^2 = Ax^2 \implies x^3 + \sqrt{A}x^2 + B = 0$. As mentioned above solving this polynomial is easy. Note that the equation $y^2 = A$ is trivially solved in $\mathbf{F}_{2^m}$, $y = A^{2^{m-1}} \implies y^2 = A$. Condition(3) $y^2 + B = 0$. Therefore point $(0, \sqrt{B})$ will always lie on the curve. But point $(0, \sqrt{y})$ has order 2. However, if point $(x, \sqrt{y})$ for any $x \in \mathbf{F}_{2^m}^*$ lies on the curve, then this point can have a large order. Such a point lies on the curve if and only if $x^2 + Ax + \sqrt{B} = 0$ has a solution in $\mathbf{F}_{2^m}$. Condition(4) $y(P) = 0$ requires to solve polynomial $x^3 + Ax^2 + B = 0$ which can be easily solved in polynomial time [5]. Condition $y([2]P) = 0$ requires to solve $\psi_3(x^2 + x + y) - (x^2 + xy) = 0$, where $\psi_3$ is a division polynomial [4,16]. Note that if $x^3 + Ax^2 + B$ has no roots, then there can be no point $(x, y)$ on the curve such that $y([2]P) = 0$.

### 6.4 Defence Against ZVP Attack Through Isomorphism For Binary Curves with $A \neq 1$

In this section we present a countermeasure against **ZVP** attack using elliptic curve isomorphism. In order to thwart the **ZVP** attack we have to choose a curve which has no **ZVP** point of large orders and is isomorphic to the original curve. Our focus will be on non-Koblitz binary curves for which $A \neq 1$ and have atleast one **ZVP** point of large order.[5] For each curve we pick a random $s$ in $\mathbf{F}_{2^m}^*$ and using isomorphism$\varphi$ we compute the corresponding curve $E'$. If $E'$ has no $\mathbf{ZVP}$ we will return $s$, otherwise we will pick another random $s$. It took us on average less than $30$ tries to find a suitable curve. Below in the table 2 we list $s$ for each **SECG** curve with $A \neq 1$. Furthermore, we will represent $s$ in hexadecimal for convenience. The conversion from hexadecimal to a field element is done by converting it to binary number and each coefficient of the binary string represents coefficients of $s$. The isomorphism computation requires only 2

[4] Koblitz curves refer to binary curves over $\mathbf{F}_{2^m}$ which have $A, B \in \{0, 1\}$.

[5] Please note that for NIST standard all non-Koblitz curves which have **ZVP** points of large orders satisfy $A = 1$. These curves are *B-163, B-283, B-571* and are discuss in next section.

field multiplications and it is much easier to store the equation of isomorphism and its inverse on memory-constraint devices (as opposed to equations of isogeny and its dual). The input points can then be mapped to the isomorphic curve for scalar multiplication and then mapped back to the original curve.

### 6.5 Isogeny Defence Against ZVP Attack for Binary Curves with $A = 1$

We recall from section 3 that over binary fields for elliptic curves with $A = 0$ or $A = 1$, **ECDBL** require 4 field multiplication instead of 5. It is clear from the definition of isomorphism that an elliptic curve $E := y^2 + xy = x^3 + Ax^2 + B$ over $\mathbf{F}_{2^m}$ with $A = 1$ and $B \neq 0, 1$ cannot be mapped to a different isomorphic curve $E' : y^2 + xy = x^3 + A^{'}x^2 + B$ with $A' = 1$. Moreover, if there exists an isomorphic curve with $A' = 0$, then the polynomial $s^2+s+1$ must have a solution in $\mathbf{F}_{2^m}$. But $s^2 + s + 1$ has no solution in $\mathbf{F}_{2^m}$, for $m = 163, 283, 571$. For these curves we need to determine the minimal isogeny degree $l_{min}$ to a curve with $A' = 0$ or $1$ and has no **ZVP** point. This would save us $l$ field multiplications. Note this is assuming that one would replace the standard curves with the corresponding isogenous curves. But if this is not a possibility, then cryptographic devices need to store along with the original curve from the standard, the isogeneous curve and the equation of the isogeny and its dual. The input points can then be mapped to the isogeneous curve for scalar multiplication and then mapped back to the original curve. Hence, we have to account for the cost of this countermeasure. The computational cost of mapping a point using isogeny of degree $l$ is $3l$ field multiplications [1,15]. Hence, in total the additional cost of isogeny defence is $6l$ field multiplications. In table 3 we list the minimal isogeny degree $l_{min}$, number of field multiplications saved by curves with $A' = 1$, number of field multiplications required by isogeny defence and the net computational cost. We observe that for curves sect283r1 (B-283) and sect571r1 (B-571) we saved 145 and 529 field multiplications, while curves sect163r2 (B-163) cost us additional 95 field multiplications. Assuming that we cannot replace standard elliptic curves with isogeneous elliptic curves, isomorphism defence is more efficient than isogeny defence for sect163r2 (B-163).

| Name Of Curve | $s$ |
|---|---|
| sect113r1 | 18FAA414E74440750490C01BB277D |
| sect113r2 | 1D15B022CC73D9E966F8A0ABFA26F |
| sect131r1 | 5EF4C6145AA39FFACD6C3296E49AB1246 |
| sect131r2 | 71C674FDCAE7A4BDE497F58E833EDF9F2 |
| sect163r1 | 5277F5AB7FFFF42D506904A46AE18086F317DFD86 |
| sect193r1 | 163E42DF9E7D5373A9C1610E3758E626CC784110B676111AD |
| sect193r2 | 1D1A0FD2202E04C7E7EF572304AE231CCBDE817720884068F |

**Table 2.** Isomorphism defence for **SECG** curves over $\mathbf{F}_{2^m}$ with $A \neq 1, B \neq 0, 1$

| Name of Curve | $l_{min}$ | Field Multiplications Saved | Cost of Isogeny | Net Cost |
|---|---|---|---|---|
| sect163r2 & B-163 | 43 | $\approx 163$, since $d \approx 2^{163}$ | 258 | 95 |
| sect283r1 & B-283 | 23 | $\approx 283$, since $d \approx 2^{283}$ | 138 | -145 |
| sect571r1 & B-571 | 7 | $\approx 571$, since $d \approx 2^{571}$ | 42 | -529 |

**Table 3.** Isogeny defence against **ZVP** points over $\mathbf{F}_{2^m}$ with $A = 1, B \neq 0, 1$

## 7 Computational Cost Comparison of Isogeny Defence with Modified Coron's 2nd and 3rd Countermeasure

We recall that Coron's 1st countermeasure (randomization of the private key $d$) and 2nd countermeasure (adding a random point to the base point ) can be modified such that they are secure against the attacks proposed in [14]. This will have an additional cost of $[(l-20)]\mathbf{A} + [(l-20)]\mathbf{D}$ and $[2(n-1)]\mathbf{A} + [2(n-1)]\mathbf{D}$ respectively, where the parameter $n$ is suggested to be $20$ [14]. The cost of **ECDBL** and **ECADD** (denoted as **A** and **D**) over $\mathbf{F}_{2^m}$ is 4 and 14 field multiplications (section 6.2). Hence for $l_{min} < 4l - 6$ the isogeny defence is faster than Coron's 1st countermeasure and for $l_{min} < 6n - 6 = 114$, the isogeny defence is faster than Coron's 2nd countermeasure. In table 4 we compare the computational cost (number of field multiplications) of the isogeny defence to these countermeasures. We note that Coron's 1st countermeasure is too expensive, that is because $l$ (size of $d$) will likely to be close to the size of the respective field size. We can see that computational cost for isogeny defense against all **ZVP** attack, is more efficient than Coron's modified countermeasures.

| Curve | Isogeny defence | Randomization of private key | Adding a random point |
|---|---|---|---|
| sect163r2 & B-163 | 222 | 2484 ($l = 158$) | 684 |
| sect283r1 & B-283 | 138 | 4644 ($l = 278$) | 684 |
| sect571r1 & B-571 | 42 | 9846 ($l = 567$) | 684 |

**Table 4.** Additional computational cost of isogeny defence vs Coron's Countermeasures over $\mathbf{F}_{2^m}$

### 7.1 Zero-Value Point Attack over $\mathbf{F}_p$

Akishita and Takagi examined the isogeny defence against **ZVP** attack over $\mathbf{F}_p$ [2]. They pointed out that most **SECG** curves have **ZVP** points from **ECDBL**. Moreover, they proved that the class of curves that satisfy $\left(\frac{-3}{p}\right) = -1$ and whose order is odd cannot be mapped by isogeny to curves with $A = -3$ and are secure against the **ZVP** attack.[6] They further point out that three **SECG** curves are in this class (secp112r1, secp192r1, secp384r1). However, they only examined **ZVP** points $3x^2 + A = 0$ and $(0, y)$ and did not consider the point $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$.[7] They also did not discuss isogeny defence against **NIST**

[6] Curves with $A = -3$ are computationally more efficient [2].

[7] These are the only **ZVP** points of large order from **ECDBL** over $\mathbf{F}_p$. See table 5.

standards and some **SECG** curves (we have place a ($\star$) before these curves, see table 5. Also in the bracket contain the isogeny degree obtain without considering all 3 **ZVP** points). We found that 5 **SECG** curves and 2 **NIST** curves contain **ZVP** points $P = (x, y)$ for which $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$. We also found that if we use an isogeny defence against all three **ZVP** points, then in some curves the isogeny degree increases dramatically. For example, for curve secp224r1, $l_m$ (minimal isogeny degree to a curve which has no **ZVP** point) and $l_p$ (minimal isogeny degree to a curve which has no **ZVP** *and* $A = -3$) increases from 1 to 3 and and 1 to 163. Note for curves which have odd order, $A = -3$ and $\left(\frac{A}{p}\right) = -1$ we list $d_p$ $\nexists$(does not exist). For each curve from the standards, we search the minimal isogeny degree $l_{min}$ to a curve which has no point $P = (x, y)$ such that $x = 0$ or $3x^2 + A = 0$ or $5x^4 + 2Ax^2 - 4bx + A^2 = 0$. If the original curve has no such point, we specify its degree 1. We also search the preferred minimal isogeny degree $l_p$ to a curve $E'$ for which $A = -3$. Again we will only discuss defence against **ZVP** points from **ECADD** since finding **ZVP** point from **ECADD** is believed to be a hard problem [2]. In tables 6 we have compared the computational cost of an isogeny defence with the modified Coron's countermeasures against the **ZVP** attack. We can see that for all curves in the computational cost for isogeny defence against the **ZVP** attack is less than Coron's modified countermeasures.

| Name of Curve | $A = -3$ | $(0, y)$ | $3x^2 + A = 0$ | $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$ | *Order* | $l_m$ | $l_p$ |
|---|---|---|---|---|---|---|---|
| secp112r1 | yes | no | yes | yes | prime | 7(7) | $\nexists$ |
| $\star$ secp112r2 | yes | yes | no | no | $4 \cdot \text{prime}$ | 13(11) | 23(11) |
| secp128r1 | yes | yes | no | no | prime | 7(7) | 181(7) |
| $\star$ secp128r2 | no | yes | no | no | $4 \cdot \text{prime}$ | 37(37) | – |
| secp160k1 | no | no | no | no | prime | 1(1) | – |
| secp160r1 | yes | yes | no | no | prime | 13(13) | 13(13) |
| secp160r2 | yes | yes | no | yes | prime | 19(19) | 227(41) |
| secp192k1 | no | no | no | no | prime | 1(1) | – |
| secp192r1 | yes | yes | yes | yes | prime | 23(23) | $\nexists$ |
| secp224k1 | no | no | no | no | prime | 1(1) | – |
| secp224r1 | yes | no | no | yes | prime | 3(1) | 163(1) |
| secp256k1 | no | no | no | no | prime | 1(1) | – |
| secp256r1 | yes | yes | no | yes | prime | 3(3) | 23(23) |
| secp384r1 | yes | yes | yes | no | prime | 31(31) | $\nexists$ |
| secp521r1 | yes | yes | yes | no | prime | 5(5) | 5(5) |
| $\star$ P-192 | yes | yes | yes | yes | prime | 23(23) | $\nexists$ |
| $\star$ P-224 | yes | no | no | yes | prime | 3(3) | 163(107) |
| $\star$ P-256 | yes | yes | no | yes | prime | 3(3) | 23(23 |
| $\star$ P-384 | yes | yes | yes | no | prime | 31(31) | $\nexists$ |
| $\star$ P-521 | yes | no | yes | yes | prime | 29(29) | $\nexists$ |

**Table 5.** list of all **SECG** and **NIST** over $\mathbf{F}_p$

## 8 Conclusion

In this paper we studied and analyzed the application of isogenies and elliptic curve isomorphisms for defence against various Zero-Value Attack. Our focus was on elliptic curve cryptosystems. We saw that these attacks can easily be thwarted by the help of elliptic curve isomorphisms (for curves over $\mathbf{F}_{2^n}$) and isogenies. These side channel attacks also point to the fact that traditional assumptions in cryptography need to be re-evaluated. Traditionally the designer of a cryptosystem assumes that an adversary knows everything about the cryptosystem being used, except the key, and has pairs of plaintext/ciphertext. However in practice more information is often available to the adversary. For example we saw that cryptographic devices leak information about private key through side channels (power consumption etc). Therefore, it is important that the cryptosystem should be designed with the assumption that unintended information is leaked by these devices. Although it's worth noting that researchers have developed hardware that leak significantly less information, so far no feasible alternatives to transistors are available. However, alternate computation technologies such as pure optical computing [8] may exist in the future. [10].

## References


1. Master's thesis.
2. T. Akishita and T. Takagi. On the Optimal Parameter Choice for EllipticCurve Cryptosystems Using Isogeny. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E88-A:140–146, 2005.
3. Toru Akishita and Tsuyoshi Takagi. Zero-value Point Attacks on Elliptic Curve Cryptosystem. In *ISC 2003, vol 2851, Lecture Notes in Computer Science (LNCS)*, pages 218–233. Springer-Verlag, 2003.
4. Ian Blake, Gadiel Seroussi, and Nigel Smart. *Elliptic Curves in Cryptography*. London Mathematical Society Lecture Note Series. Cambridge University press, 2006.
5. Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, 1993.
6. Jean-Sébastien Coron. Resistance.
7. Taher ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, 31 (4):469–472, 1985.
8. Louis Goubin. A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography*, volume Lecture Notes in Computer Science 2567, pages 199–210. Springer-Verlag, 2003.
9. Marc Joye and Christophe Tymen. Protections against Differential Analysis for Elliptic Curve Cryptography. In *Cryptographic Hardware and Embedded Systems-CHES (2001)*, volume Lecture Notes in Computer Science 2162, pages 377–390. Springer-Verlag, 2001.
10. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Introduction to differential power analysis and related attacks. Technical report, Cryptography Research, 1998.
11. Paul C. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *CRYPTO '96 Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, volume 1109, pages 104–113. Springer-Verlag, 1996.
12. J. Lopez and R. Dahab. Fast Multiplication on Elliptic Curves Over $GF(2^m)$ Without precomputation. In *Cryptographic Hardware and Embedded Systems-CHES(99)*, volume 1717, pages 292–302. Springer-Verlag, 1999.


[8] An optical computer is a computer that uses light instead of electricity to manipulate, store and transmit data

13. Alfred J. Menezes. *Elliptic Curve Public Key Cryptosystems*. Springer, 1993.
14. Katsuyuki Okeya and Kouichi sakurai. Power Analysis Breaks Elliptic Curve Cryptosystems Even Secure against the Timing Attack. In *Progress in Cryptology–INDOCRYPT 2000*, volume 1977, pages 178–190. Springer-Verlag, 2000.
15. Nigel P. Smart. An Analysis of Goubin's Refined Power Analysis Attack. In *Cryptographic Hardware and Embedded Systems-CHES (2003)*, volume Lecture Notes in Computer Science 2779, pages 281–290. Springer Berlin / Heidelberg, 2003.
16. Lawrence C. Washington. *Elliptic Curves Number Theory and Cryptography*. Chapman and Hall, 2003.

## A Implementation of Elliptic Curve Point Doubling (ECDBL) and Point Addition (ECADD) over $\mathbf{F}_2^m$

If we use algorithm 1 for scalar multiplication then for over $\mathbf{F}_{2^m}$ the most efficient method of point doubling and point addition was proposed in [12]. In their, paper affine coordinates $(x, y)$ were mapped to projective coordinate $(X, Y, Z)$ by setting $x = X/Z$ and $y = Y/Z^2$. The equation of elliptic curves over these projective coordinates is given by $E : Y^2 + XYZ = X^3Z + A(XZ)^2 + BZ^4$. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points on $E$.
**Inverse and Point of infinity:** $-P_1 = (X_1, X_1Z_1 + Y_1, Z_1)$ and $P_\infty = [(\alpha, 0, 0)]$ for any $\alpha \in \mathbf{F}_{2^m}$ and $\alpha \neq 0$.

**ECDBL** : $[2]P_1 = (X, Y, Z)$, $Z = Z_1^2 \cdot X_1^2$, $X = X_1^4 + BZ_1^4$, $Y = BZ_1^4 \cdot Z + X \cdot (AZ + Y_1^2 + BZ_1^4)$.

**ECADD** : $P_1 + P_2 = (X_3, Y_3, Z_3)$, $X_3 = C^2 + H + G$, $Y_3 = HI + Z_3J$, $Z_3 = F^2$.

Where, $A_0 = Y_2Z_1^2$, $A_1 = Y_1Z_2^2$, $B_0 = X_2Z_1$, $B_1 = X_1Z_2$, $C = A_0 + A_1$, $D = B_0 + B_1$, $E = Z_1Z_2$, $F = DE$, $G = D^2(F + AE^2)$, $H = CF$, $I = D^2B_0E + X_3$, $J = D^2A_0 + X_3$.

ECDBL required 5 field multiplications in general and 4 multiplications if $A = 0$ or $A = 1$. ECADD required 14 field multiplications in general and 9 multiplications if $A = 0$ or $A = 1$ and $Z = 1$.

- *Implementation of Elliptic Curve Point Doubling* **ECDBL** *for Binary Fields*
- **Input** $(P_1 \neq P_\infty, A, c = B^{2^{m-1}})$
- **Output** $(2P_1)$

1. $T_1 \longleftarrow X_1$ , $T_2 \longleftarrow Y_1$ , $T_3 \longleftarrow Z_1$
2. $T_4 \longleftarrow c$
3. $T_3 \longleftarrow T_3 \times T_3$ : $(= Z_1^2)$
4. $T_4 \longleftarrow T_3 \times T_4$ : $(= cZ_1^2)$
5. $T_4 \longleftarrow T_4 \times T_4$ : $(= BZ_1^4)$
6. $T_1 \longleftarrow T_1 \times T_1$ : $(= X_1^2)$
7. $T_3 \longleftarrow T_1 \times T_3$ : $(= X_1^2Z_1^2 = Z_2)$
8. $T_1 \longleftarrow T_1 \times T_1$ : $(= X_1^4)$
9. $T_1 \longleftarrow T_1 + T_4$ : $(= X_1^4 + BZ_1^4 = X_2)$
10. $T_2 \longleftarrow T_2 \times T_2$ : $(= Y_1^2)$
11. If $A \neq 0$
    $T_5 \longleftarrow A$ :
    $T_5 \longleftarrow T_3 \times T_5$ :
    $T_2 \longleftarrow T_5 + T_2$ : $(= AZ_2 + Y_1^2)$

12. $T_2 \longleftarrow T_2 + T_4$ : $(= AZ_2 + Y_1^2 + BZ_1^4)$ *or* $(= Y_1^2 + BZ_1^4)$
13. $T_2 \longleftarrow T_1 \times T_2$ : $(= X_2(AZ_2 + Y_1^2 + BZ_1^4))$ *or* $(= X_2(Y_1^2 + BZ_1^4))$
14. $T_4 \longleftarrow T_3 \times T_4$ : $(= BZ_2Z_1^4)$
15. $T_2 \longleftarrow T_2 + T_4$ : $(= BZ_2Z_1^4 + X_2(Y_1^2 + BZ_1^4) = Y_2)$ *or* $(= BZ_2Z_1^4 + X_2(AZ_2 + Y_1^2 + BZ_1^4) = Y_2)$
16. $X_2 \longleftarrow T_1$
17. $Y_2 \longleftarrow T_2$
18. $Z_2 \longleftarrow T_3$

– *Implementation of Elliptic Curve Point Addition* **ECADD** *for Binary Fields*
– **Input** $(P_1 \neq P_\infty,\ P_2 \neq P_\infty,\ A,\ B)$
– **Output** $(P_1 + P_2)$

1. $T_1 \longleftarrow X_1,\ T_2 \longleftarrow Y_1,\ T_3 \longleftarrow Z_1$
2. $T_4 \longleftarrow X_2,\ T_5 \longleftarrow Y_2,\ T_6 \longleftarrow Z_2$
3. $T_7 \longleftarrow T_4 \times T_3$ : $(= X_2Z_1 = B_0)$
4. $T_1 \longleftarrow T_6 \times T_1$ : $(= X_1Z_2 = B_1)$
5. $T_8 \longleftarrow T_3 \times T_6$ : $(= Z_1Z_2 = E)$
6. $T_3 \longleftarrow T_5 \times T_7$ : $(= Y_2Z_1^2 = A_0)$
7. $T_6 \longleftarrow T_6 \times T_6$ : $(= Z_2^2)$
8. $T_6 \longleftarrow T_2 \times T_6$ : $(= Y_1Z_2^2 = A_1)$
9. $T_2 \longleftarrow T_3 + T_6$ :: $(= A_0 + A_1 = Y_2Z_1^2 + Y_1Z_2^2 = C)$
10. $T_4 \longleftarrow T_1 + T_7$ : $(= B_0 + B_1 = X_2Z_1 + X_1Z_2 = D)$
11. $T_5 \longleftarrow T_4 \times T_8$ : $(= D(Z_1Z_2) = F)$
12. $T_6 \longleftarrow T_5 \times T_5$ : $(= F^2 = Z_3)$
13. $T_4 \longleftarrow T_4 \times T_4$ : $(= D^2)$
14. $T_9 \longleftarrow T_8 \times T_8$ : $(= E^2)$
15. $T_9 \longleftarrow A \times T_9$ : $(= AE^2)$
16. $T_9 \longleftarrow T_5 + T_9$ : $(= F + AE^2)$
17. $T_9 \longleftarrow T_4 \times T_9$ : $(= (D^2)(F + AE^2) = G)$
18. $T_1 \longleftarrow T_2 \times T_2$ : $(= (Y_2Z_1)^2 + (Y_1Z_2)^2 = C^2)$
19. $T_2 \longleftarrow T_2 \times T_5$ : $(= CF = H)$
20. $T_1 \longleftarrow T_1 + T_2$ : $(= C^2 + H)$
21. $T_1 \longleftarrow T_1 + T_9$ : $(C^2 + H + G = X_3)$
22. $T_5 \longleftarrow T_4 \times T_7$ : $(= D^2B_0)$
23. $T_5 \longleftarrow T_5 \times T_8$ : $(= B_0D^2E)$
24. $T_5 \longleftarrow T_5 + X_2$ : $(= B_0D^2E + X_3 = I)$
25. $T_8 \longleftarrow T_4 + T_3$ : $(= A_0D^2)$
26. $T_8 \longleftarrow T_8 + T_2$ : $(= A_0D^2 + X_3 = J)$
27. $T_8 \longleftarrow T_6 \times T_8$ : $(= Z_3J)$
28. $T_2 \longleftarrow T_2 \times T_5$ : $(= HI)$
29. $T_2 \longleftarrow T_2 + T_8$ : $(= HI + Z_3J = Y_3)$

## B Implementation of Elliptic Curve Point Doubling (ECDBL) and Point Addition (ECADD) $\mathbf{F}_p$

For prime fields we use Jacobian projective coordinates. In this system, affine coordinates $(x, y)$ were mapped to projective coordinate $(X, Y, Z)$ by setting $x = X/Z^2$ and $y = Y/Z^3$. The

equation of elliptic curves over Jacobian coordinates is given by $E : Y^2 = X^3 + AXZ^4 + BZ^6$. Let $P_1 = (X_1, Y_1, Z_1)$ and $P_2 = (X_2, Y_2, Z_2)$ be two points on $E$.

**Inverse and Point of infinity:** $-P_1 = (X_1, -Y_1, Z_1)$ and $P_\infty = [(0, \alpha, 0)]$ for any non-zero $\alpha \in \mathbf{P^2}_{\mathbf{F}_p}$.

**ECDBL**: $X_3 = T$, $Y_3 = -8Y_1^4 + M(S - T)$, $Z_3 = 2Y_1Z_1$.

**ECADD**: $X_3 = -H^3 - 2U_1H^2 + R^2$, $Y_3 = -S_1H^3 + R(U_1H^2 - X_3)$, $Z_3 = Z_1Z_2H$

Where, $U_1 = X_1Z_2^2$, $U_2 = X_2Z_1^2$, $S_1 = Y_1Z_2^3$, $S_2 = Y_2Z_1^3$, $H = U_2 - U_1$, $R = S_2 - S_1$, $S = 4X_1Y_1^2$, $M = 3X_1^2 + AZ_1$, $T = -2S + M^2$

ECDBL require 10 multiplications. ECADD require 16 multiplications in general and 8 if $A = -3$.

- *Implementation of Elliptic Curve Point Double* **ECDBL** *for Primary Fields*
- **Input** $(P_1 \neq P_\infty, A)$.
- **Output** $(2P_1)$.
  1. $T_4 \longleftarrow X_1$ , $T_5 \longleftarrow Y_1$ , $T_6 \longleftarrow Z_1$
  2. $T_1 \longleftarrow T_4 \times T_4$ : $(= X_1^2)$
  3. $T_2 \longleftarrow T_5 \times T_5$ : $(= Y_1^2)$
  4. $T_2 \longleftarrow T_2 + T_2$ : $(= 2Y_1^2)$
  5. $T_4 \longleftarrow T_4 \times T_2$ : $(= 2X_1Y_1^2)$
  6. $T_4 \longleftarrow T_4 + T_4$ : $(= 4X_1Y_1^2 = S)$
  7. $T_2 \longleftarrow T_2 \times T_2$ : $(= 4Y_1^4)$
  8. $T_2 \longleftarrow T_2 + T_2$ : $(= 8Y_1^4)$
  9. $T_3 \longleftarrow T_6 \times T_6$ : $(= Z_1^2)$
  10. $T_3 \longleftarrow T_3 \times T_3$ : $(= Z_1^4)$
  11. $T_6 \longleftarrow T_5 \times T_6$ : $(= Y_1Z_1)$
  12. $T_6 \longleftarrow T_6 + T_6$ : $(= 2Y_1Z_1)$
  13. $T_5 \longleftarrow T_1 + T_1$ : $(= 2X_1^2)$
  14. $T_1 \longleftarrow T_1 + T_5$ : $(= 3X_1^2)$
  15. $T_3 \longleftarrow A \times T_3$ : $(= AZ_1^4)$
  16. $T_1 \longleftarrow T_1 + T_3$ : $(= 3X_1^2 + AZ_1^4 = M)$
  17. $T_3 \longleftarrow T_1 \times T_1$ : $(= M^2)$
  18. $T_3 \longleftarrow T_3 - T_4$ : $(= M^2 - S)$
  19. $T_3 \longleftarrow T_3 - T_4$ : $(X_3 = M^2 - 2S = T)$
  20. $T_4 \longleftarrow T_4 - T_3$ : $(= S - T)$
  21. $T_1 \longleftarrow T_1 \times T_4$ : $(= M(S - T))$
  22. $T_4 \longleftarrow T_1 - T_2$ : $(= 8Y_1^4 - M(S - T))$
  23. $X_3 \longleftarrow T_3$ , $Y_3 \longleftarrow T_4$ , $Z_3 \longleftarrow T_6$

- *Implementation of Elliptic Curve Point Addition* $\mathbf{ECADD}$ *for Primary Fields*
- **Input** $(P_1 \neq P_\infty, P_2 \neq P_\infty)$.
- **Output** $(P_3)$.
  1. $T_2 \longleftarrow X_1$ , $T_3 \longleftarrow Y_1$ , $T_4 \longleftarrow Z_1$
  2. $T_5 \longleftarrow X_2$ , $T_6 \longleftarrow Y_2$ , $T_7 \longleftarrow Z_2$
  3. $T_1 \longleftarrow T_7 \times T_7$ : $(= Z_2^2)$
  4. $T_2 \longleftarrow T_2 \times T_1$ : $(= X_1Z_2^2 = U_1)$
  5. $T_3 \longleftarrow T_3 \times T_7$ : $(= Y_1Z_2)$

6. $T_3 \longleftarrow T_3 \times T_1 : \quad (= Y_1 Z_2^3 = S_1)$
7. $T_1 \longleftarrow T_4 \times T_4 : \quad (= Z_1^2)$
8. $T_5 \longleftarrow T_5 \times T_1 : \quad (= X_2 Z_1^2 = U_2)$
9. $T_6 \longleftarrow T_6 \times T_4 : \quad (= Y_2 Z_1)$
10. $T_6 \longleftarrow T_6 \times T_1 : \quad (= Y_2 Z_1^3 = S_2)$
11. $T_5 \longleftarrow T_5 - T_2 : \quad (= U_2 - U_1 = H)$
12. $T_7 \longleftarrow T_4 \times T_7 : \quad (= Z_1 Z_2)$
13. $T_7 \longleftarrow T_5 \times T_7 : \quad (= Z_1 Z_2 H = Z_3)$
14. $T_6 \longleftarrow T_6 - T_3 : \quad (= S_2 - S_1 = R)$
15. $T_1 \longleftarrow T_5 \times T_5 : \quad (= H^2)$
16. $T_4 \longleftarrow T_6 \times T_6 : \quad (= R^2)$
17. $T_2 \longleftarrow T_2 \times T_1 : \quad (= U_1 H^2)$
18. $T_5 \longleftarrow T_5 \times T_1 : \quad (= H^3)$
19. $T_4 \longleftarrow T_4 - T_5 : \quad (= R^2 - H^3)$
20. $T_1 \longleftarrow T_2 + T_2 : \quad (= 2U_1 H^2)$
21. $T_4 \longleftarrow T_4 - T_1 : \quad (= -H^3 - 2U_1 H^2 + R^2 = X_3)$
22. $T_2 \longleftarrow T_2 - T_4 : \quad (= U_1 H^2 - X_3)$
23. $T_6 \longleftarrow T_6 \times T_2 : \quad (= R(U_1 H^2 - X_3))$
24. $T_1 \longleftarrow T_3 \times T_5 : \quad (= S_1 H^3)$
25. $T_1 \longleftarrow T_6 - T_1 : \quad (= S_1 H^3 + R(U_1 H^2 - X_3))$
26. $X_3 \longleftarrow T_4$, $Y_3 \longleftarrow T_1$, $Z_3 \longleftarrow T_7$

# C Zero-Value Points over $\mathbf{F}_p$

**Theorem 2.** Let $E : y^2 = x^3 + Ax + B$ be an elliptic curve over prime field $\mathbf{F}_p$. The elliptic curve $E$ has a zero value point $P = (x, y)$ of **ECADD** if and only if one of the following conditions are satisfied: $(1)\, 3x^2 + A = 0,\ (2)\, 5x^4 + 2Ax^2 - 4Bx + A^2 = 0,\ (3)\, [3]P = P_\infty,\ (4)\, x(P) = 0$ or $x([2]P) = 0$ *or* $,(5)\, y(P) = 0$ or $y([2]P) = 0$. Moreover, the zero-value points are not randomized by the three countermeasures (randomized projective coordinates, random elliptic curve isomorphism, random field isomorphism).

*Proof.* [3]

## C.1 Finding Zero-Value Point from ECDBL

We recall from see section 5.3 that points of small order can be dealt with by careful implementation of scalar multiplication algorithm. Hence, we do not have to worry about condition (3), and condition (5). In char$> 3$ $(x, 0)$ has order 2 and $(x, y)$, such that $x([2](x, y) = 0$ has order 4. For condition (1) we have to solve the polynomial $3x^2 + A = 0$ and for condition (2) we have to solve the polynomial $5x^4 + 2Ax^2 - 4Bx + A^2 = 0$ and for condition. For condition (4) we have to solve the polynomial $y^2 - B = 0$. The solutions for these polynomials over finite fields can be easily computed in polynomial time, for details see [5].

## D Tables

| Curve | #multi for $l_m$ | #multi for $l_p$ | #multi randomization | #multi blinding |
|---|---|---|---|---|
| secp112r1 | 42 | ∄ | 442 ($l = 37$) | 988 |
| ⋆secp112r2 | 78 | 138 | 2088 ($l = 107$) | 912 |
| secp128r1 | 42 | 1086 | 2472 ($l = 123$) | 912 |
| ⋆secp128r2 | 222 | N/A | 2678 ($l = 123$) | 988 |
| secp160r1 | 78 | 78 | 3240 ($l = 155$) | 912 |
| secp160r2 | 114 | 1362 | 3240 ($l = 155$) | 912 |
| secp192r1 | 138 | ∄ | 4342 ($l = 187$) | 988 |
| secp224r1 | 18 | 978 | 4776 ($l = 199$) | 912 |
| secp256r1 | 18 | 138 | 5544 ($l = 251$) | 912 |
| secp384r1 | 186 | ∄ | 9334 ($l = 379$) | 988 |
| secp521r1 | 30 | 30 | 11904 ($l = 516$) | 912 |
| P-192 | 138 | ∄ | 4342 ($l = 187$) | 988 |
| P-224 | 18 | 978 | 4776 ($l = 199$) | 912 |
| P-256 | 18 | 138 | 5544 ($l = 251$) | 912 |
| P-384 | 186 | ∄ | 9334 ($l = 379$) | 988 |
| P-521 | 174 | ∄ | 12896 ($l = 516$) | 988 |

**Table 6.** Comparison of additional computational cost for **SECG** and **NIST** curves over $\mathbf{F}_p$