Exceptional Procedure Attack on Elliptic Curve Cryptosystems

Tetsuya Izu[1] and Tsuyoshi Takagi[2]

[1] FUJITSU LABORATORIES Ltd.
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan
izu@flab.fujitsu.co.jp
[2] Technische Universität Darmstadt, Fachbereich Informatik
Alexanderstr.10, D-64283 Darmstadt, Germany
ttakagi@cdc.informatik.tu-darmstadt.de

Abstract. The scalar multiplication of elliptic curve based cryptosystems (ECC) is computed by repeatedly calling the addition formula that calculates the elliptic curve addition of two points. The addition formula involves several exceptional procedures so that implementers have to carefully consider their treatments. In this paper we study the exceptional procedure attack, which reveals the secret scalar using the error arisen from the exceptional procedures. Recently new forms of elliptic curves and addition formulas for ECC have been proposed, namely the Montgomery form, the Jacobi form, the Hessian form, and the Brier-Joye addition formula. They aim at improving security or efficiency of the underlying scalar multiplications. We analyze the effectiveness of the exceptional procedure attack to some addition formulas. We conclude that the exceptional procedure attack is infeasible against the curves whose order are prime, i.e., the recommended curves by several standards. However, the exceptional procedure attack on the Brier-Joye addition formula is feasible, because it yields non-standard exceptional points. We propose an attack that reveals a few bits of the secret scalar, provided that this multiplier is constant and fixed. By the experiment over the standard elliptic curves, we have found many non-standard exceptional points even though the standard addition formula over the curves has no exceptional point. When a new addition formula is developed, we should be cautious about the proposed attack.

Keywords: Elliptic curve cryptosystem (ECC), scalar multiplication, exceptional procedure attack, exceptional point, side channel attack

1 Introduction

The scalar multiplication of the elliptic curve cryptosystem (ECC) is implemented using the addition formula assembled by arithmetics of a definition field. The addition formula involves exceptional procedures that cause an error (0^{-1} of the definition field) at the end of the scalar multiplication. Implementers should carefully deal with the exceptional procedures. Recently new forms of elliptic curves and addition formulas have been proposed, namely the Montgomery

Y.G. Desmedt (Ed.): PKC 2003, LNCS 2567, pp. 224–239, 2003.
© Springer-Verlag Berlin Heidelberg 2003

form [OKS00], the Jacobi form [LS01, BiJ02], the Hessian form [JQ01, Sma01], and the Brier-Joye addition formula [BrJ02]. These new curves and formulas aim at improving security or efficiency of their scalar multiplications.

In this paper we study a possible attack using the error arisen from the exceptional procedures of the addition formula, called the *exceptional procedure attack*. Two points are called the exceptional points if they cause the exceptional procedure. The goal of the attack is to reveal the secret key d. The attack tries to produce the exceptional point of the elliptic curve E during the scalar multiplication by manipulating the base point $P \in E$ to the appropriate point $P' \in E$. If an error occurs in the scalar multiplication $d * P' \in E$, the attack can know a few significant bits of d depending on the underlying addition chain. A basic principal of the proposed attack is different from that of the fault based attacks [BDL97, BMM00] or the small subgroup attack [LMQSV98]. The fault attacks usually analyze the scalar multiplication $d * P^* \in E^*$ with the point P^* over the different curve E^*, which need a physical fault in order to generate a failure point P^*. The small subgroup attack performs a calculation of $d*P$ over the subgroup $\mathrm{Tor}(E) \subset E$ with small order, and the order of the elliptic curve must be divisible by a small integer. The exceptional procedure attack essentially requires neither the physical fault nor the curves with the small subgroup.

We first discuss the exceptional procedure attack against the standard addition formula. We observe necessary and sufficient conditions that two points are exceptional points, that is P_1 or P_2 is contained in the torsion subgroups of the underlying group. When the order of the elliptic curve is divisible by a small integer, the curve has several exceptional points. In other words, the curves whose order are prime are secure against the proposed attack, namely the recommended curves in several standards [ANSI, IEEE, SEC].

Next we analyze the exceptional procedure attack against the non-standard addition formula proposed by Brier-Joye [BrJ02]. This addition formula is designed for enhancing security of the scalar multiplication, namely resistant against the side channel attack [KJJ99]. Their exceptional points are two points $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$, which only satisfy $y_1 + y_2 = 0$. Some of these points are non-trivial exceptional points. Especially, a point P is called the m-th collision point if two points mP, P are exceptional points. We show necessary and sufficient conditions that a point P is the m-th collision point. We demonstrate that there exist many m-th collision points over the recommended curves by the international standards [ANSI, IEEE, SEC]. Moreover, we analyze a possible attack against the plain ElGamal cryptosystem using the collision points.

The attack proposed in this paper is independent from the characteristic of the definition field. However, just for a simplicity, we only discuss prime field cases in the following.

2 Elliptic Curves

In this section we briefly review basic properties of elliptic curves. The standard addition formula and its variants over different coordinate systems are described. We then explain how the scalar multiplication provides an error.

2.1 Standard Addition Formula

Let $\mathbb{F}_p$ be a finite field with p elements, where $p > 3$ is a prime. Let E be an elliptic curve over $\mathbb{F}_p$ defined by Weierstrass-form equation

$$E : y^2 = x^3 + a\ x + b \quad (a, b \in \mathbb{F}_p,\ 4a^3 + 27b^2 \neq 0). \tag{1}$$

A point of E is uniquely represented as (x, y) that is the pair in the basic field $\mathbb{F}_p$. It is called the affine coordinate representation. A set of all points on curve E, including the point of infinity $\mathcal{O}$, is denoted by $E(\mathbb{F}_p)$. This set has a commutative additive group structure with the neutral element $\mathcal{O}$. We denote by $+$ the addition of the $E(\mathbb{F}_p)$. Especially we have $P+\mathcal{O} = P$ and $P+(-P) = \mathcal{O}$, where $-P = (x, -y)$ for point $P = (x, y) \in E(\mathbb{F}_p) \setminus \{\mathcal{O}\}$.

In order to algorithmically describe the addition rule arisen from the addition group $E(\mathbb{F}_p)$, we employ the arithmetic of the definition field $\mathbb{F}_p$, namely additions, subtractions, multiplications, and inversions. The addition rule is called the addition formula.

We explain the standard addition formula in the following. Let $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ be points on $E(\mathbb{F}_p)$ that are different from $\mathcal{O}$. The standard addition formula calculates the point $P_3 = (x_3, y_3)$ of the addition $P_3 = P_1 + P_2$. The standard addition formula is as follows:

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2, \\ y_3 = -\lambda x_3 - \mu, \end{cases}$$

where

$$(\lambda, \mu) = \begin{cases} \left(\frac{y_2 - y_1}{x_2 - x_1}, \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}\right), & x_1 \neq x_2 \\ \left(\frac{3x_1^2 + a}{2y_1}, \frac{-x_1^3 + a\ x_1 + 2b}{2y_1}\right), & (x_1 = x_2) \wedge (P_2 \neq -P_1) \end{cases}$$

We remark that there are two formulas for (λ, μ) based on the condition of two points P_1, P_2. The first one is used for the case $P_1 \neq P_2$ and it is called ECADD. The second one is used for the case $P_1 = P_2$ and is called ECDBL. Note that the standard addition formula does not support the points P_1, P_2 with $P_1 + P_2 = \mathcal{O}$, $P_1 = \mathcal{O}$, $P_2 = \mathcal{O}$ for ECADD and $P = (x, 0)$ for ECDBL. It is called the exceptional procedure of the standard addition formula, because they require exceptional treatment in the formula.

2.2 Coordinate System

The standard addition formula described in the previous section is designed for the affine representation of points. It possesses one inversion of the definition

field, which is relatively expensive in most computing environments. Fortunately the elliptic curve has different coordinate systems which do not need inversions. The projective coordinate and the Jacobian coordinate are examples [CMO98]. Using these coordinates we can avoid computing inversions in $\mathbb{F}_p$. In the following we represent the standard addition formula using the projective coordinate and the Jacobian coordinate, which are widely employed in elliptic curve cryptosystems.

In the projective coordinate, a point is represented by a tuple $(X : Y : Z)$, where two points $(X : Y : Z)$ and $(rX : rY : rZ)$ $(r \in \mathbb{F}_p^*)$ are identified as the same point. The curve equation is given by substituting $x = X/Z,\ y = Y/Z$ into (1). The identity point $\mathcal{O}$ is represented by $(0 : 1 : 0)$; this is the only point with its Z-coordinate equal to 0. Setting $x = X/Z,\ y = Y/Z$ in the affine equation, we obtain the projective Weierstrass equation $E_{\mathcal{P}} : Y^2Z = X^3 + a\ XZ^2 + bZ^3$. The inverse of $P = (X : Y : Z)$ is $-P = (X : -Y : Z)$. Let $P_1 = (X_1 : Y_1 : Z_1),\ P_2 = (X_2 : Y_2 : Z_2)$ and $P_1 + P_2 = P_3 = (X_3 : Y_3 : Z_3)$, then the addition formulas are as follows:

ECADD in Projective Coordinate $(P_1 \neq \pm P_2)$: $X_3 = vA,\ Y_3 = u(v^2X_1Z_2 - A) - v^3Y_1Z_2,\ Z_3 = v^3Z_1Z_2$ with $u = Y_2Z_1 - Y_1Z_2,\ v = X_2Z_1 - X_1Z_2,\ A = u^2Z_1Z_2 - v^3 - 2v^2X_1Z_2$

ECDBL in Projective Coordinate $(P_1 = P_2)$: $X_3 = 2hs,\ \ Y_3 = w(4B - h) - 8Y_1^2s^2,\ \ Z_3 = 8s^3$ with $w = aZ_1^2 + 3X_1^2,\ s = Y_1Z_1,\ B = X_1Y_1s,\ h = w^2 - 8B$

If $Z_1 = 0$, then $Z_3 = 0$ for both ECADD and ECDBL. Thus the point $P = (X : Y : 0)$ is the exceptional points of the standard addition formula using the projective coordinate.

The Jacobian coordinate system offers a faster computation of the addition formula. In this coordinate, a point on the curve is represented as a tuple $(X : Y : Z)$. Two points $(X : Y : Z)$ and $(r^2X : r^3Y : rZ)$ $(r \in \mathbb{F}_p^*)$ are identified as the same point. The identity point $\mathcal{O}$ is represented by $(0 : 1 : 0)$; this is again the only point with its Z-coordinate equal to 0. Setting $x = X/Z^2,\ y = Y/Z^3$ in the affine equation, we obtain the Jacobian equation $E_{\mathcal{J}} : Y^2 = X^3 + a\ XZ^4 + bZ^6$. The inverse of $P = (X : Y : Z)$ is $-P = (X : -Y : Z)$. Let $P_1 = (X_1 : Y_1 : Z_1),\ P_2 = (X_2 : Y_2 : Z_2)$ and $P_1 + P_2 = P_3 = (X_3 : Y_3 : Z_3)$, then the addition formulas are as follows:

ECADD in Jacobian Coordinate $(P_1 \neq \pm P_2)$: $X_3 = -H^3 - 2U_1H^2 + r^2,\ Y_3 = -S_1H^3 + r(U_1H^2 - X_3),\ Z_3 = Z_1Z_2H$ with $U_1 = X_1Z_2^2,\ U_2 = X_2Z_1^2,\ S_1 = Y_1Z_2^3,\ S_2 = Y_2Z_1^3,\ H = U_2 - U_1,\ r = S_2 - S_1$

ECDBL in Jacobian Coordinate $(P_1 = P_2)$: $X_3 = T,\ Y_3 = -8Y_1^4 + M(S - T),\ Z_3 = 2Y_1Z_1$ with $S = 4X_1Y_1^2,\ M = 3X_1^2 + aZ_1^4,\ T = -2S + M^2$

As we discuss for the projective coordinate, if $Z_1 = 0$, then $Z_3 = 0$ for both ECDBL and ECDBL. Thus the point $P_1 = (X : Y : 0)$ is the exceptional points of the standard addition formula using the Jacobian coordinate.

2.3 Scalar Multiplication

Let d be an n-bit integer and P be a point on the elliptic curve $E(\mathbb{F}_p)$. The scalar multiplication is to compute the point $d * P = P + P + \cdots + P$ ($d - 1$ additions). This is the dominant computation of all ECC algorithms, including the encryption/decryption and the signature generation/verification.

The standard algorithm for computing the scalar multiplication is the binary method. Let $d = d[0]2^0 + d[1]2^1 + ... + d[n-1]2^{n-1}$ be the binary representation of the scalar d, where $d[n-1]$ is the most significant bit of d and $d[n-1] = 1$. Then the binary addition chain computes the scalar multiplication $d*P$ for given $d[0], d[1], ..., d[n-1]$ and point P as follows. We first assign $T = P$. For $i = n-2$ down to 0 we compute $T = \mathrm{ECDBL}(T)$ and $T = \mathrm{ECADD}(T, P)$ if $d[i] = 1$. Finally T is returned as the value of the scalar multiplication $d * P$.

With the projective coordinate, a scalar multiplication $d * P = (x_d, y_d)$ of $P = (x, y)$ is computed as follows:

1. Set $(X : Y : Z) = (x : y : 1)$.
2. Compute $(X_d : Y_d : Z_d) = d * (X : Y : Z)$.
3. Convert $(X_d : Y_d : Z_d)$ to $(x_d, y_d) = (X_d/Z_d, Y_d/Z_d)$.

Note that if $Z_d = 0$ in Step 3, the conversion fails. Similarly, the conversion fails using the Jacobian coordinate if the Z-coordinate is zero in Step 3. Once the Z-coordinate of the projective (or Jacobian) coordinate becomes zero during the scalar multiplication, the error will be occurred in Step 3. The error is usually returned as the system error and we can observe that the exceptional procedure of the addition formula has caused during the scalar multiplication.

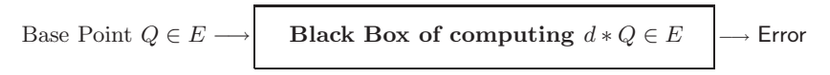
3 Exceptional Procedure Attack

In this section, we propose the exceptional procedure attack by using the exceptional procedure in the addition formula and we analyze its effectiveness for the standard addition formula. This section aims at proposing the general idea of the exceptional procedure attack using the standard addition formula. The analysis of this attack against the other addition formulas (or the addition formula for hyper-elliptic curves) strongly depend on their explicit formula. Details of the analysis for each addition formula must be independently considered. Indeed we deeply analyze the Brier-Joye addition formula in the next section.

3.1 Basic Idea

Let P be a base point of an elliptic curve E and d be a secret scalar. The exceptional procedure attack tries to reveal (part of) the secret key d. An idea of the attack is to produce the exceptional point over E, which causes an error ($0^{-1} \in \mathbb{F}_p$) at the end of the scalar multiplication. The secret key d is guessed from the error of the scalar multiplication $d * Q$ for different base points Q of the curve E. For example the replacement can be accomplished by the chosen

ciphertext attack. The attacker uses the scalar multiplication $d * Q$ for chosen point Q as a black box.



In order to achieve this scenario, we assume the following two assumptions for our attack setting.

1. (Base Point Replacement) In the beginning of the scalar multiplication $d * P \in E$, the attacker can replace the base point P to another point Q of the elliptic curve E.
2. (Error Detection) The attacker can detect the error caused by the final inversion ($0^{-1} \in \mathbb{F}_p$) of the scalar multiplication.

Instead of outputting the error ($0^{-1} \in \mathbb{F}_p$), one can return 0 (or some other value) for Z-coordinate. However, one can still detect the error, because the returned point is not a correct value of the scalar multiplication $d * Q \in E$ and thus it causes an error of the cryptographic primitive in the decryption process.

One of the main theme of our attack is how to produce these exceptional points. We first investigate the occurrence criteria of exceptional points.

3.2 Exceptional Procedure in Standard Formula

We investigate conditions with which the standard formula has the exceptional procedures, namely the Z-coordinate of the addition $P_1 + P_2$ becomes zero. We consider the standard formula using the projective coordinate and the Jacobian coordinate.

First, we look at the standard formula using the projective coordinate. Let $P_1 = (X_1 : Y_1 : Z_1)$, $P_2 = (X_2 : Y_2 : Z_2)$ with $P_1 \neq P_2$. Then, from the standard formula using the projective coordinate, the Z-coordinate of the addition $P_3 = (X_3 : Y_3 : Z_3) = P_1 + P_2$ is computed by $Z_3 = v^3 Z_1 Z_2$ for $v = X_2 Z_1 - X_1 Z_2$. If $Z_3 = 0$, we have three cases, (1) $v = 0$, (2) $Z_1 = 0$, or (3) $Z_2 = 0$. Suppose $v = X_2 Z_1 - X_1 Z_2 = 0$. If $(X_2 = 0) \wedge (X_1 = 0)$, $P_1 = \pm P_2$. If $(X_2 = 0) \wedge (Z_2 = 0)$, $P_2 = \mathcal{O}$. If $(Z_1 = 0) \wedge (X_1 = 0)$, $P_1 = \mathcal{O}$. If $(Z_1 = 0) \wedge (Z_2 = 0)$, $P_1 = P_2 = \mathcal{O}$. Suppose $v \neq 0$. Then we have $P_1 = \mathcal{O}$ $(Z_1 = 0)$ or $P_2 = \mathcal{O}$ $(Z_2 = 0)$. These observations are summarized as follows: $Z_3 = 0$ iff $P_1 = \pm P_2$ or (at least) one of P_1, P_2 is $\mathcal{O}$. These points coincide the exceptional points in the standard addition formula except $P_1 = P_2$. If $P_1 = P_2$, we use the formula ECDBL. The Z-coordinate of ECDBL(P_1) becomes zero iff $Y_1 = 0$ or $Z_1 = 0$ holds. We can compute all points P with $Y_1 = 0$ using the definition equation $x^3 + ax + b = 0$. This equation has solutions over $\mathbb{F}_p$ iff the order of the curve is divisible by 2.

Next we consider the Jacobian case. Let Z_3 be the Z-coordinate of the addition $P_1 + P_2$ using the Jacobian coordinate, where $P_1 = (X_1 : Y_1 : Z_1)$, $P_2 = (X_2 : Y_2 : Z_2)$ with $P_1 \neq P_2$. We have $Z_3 = Z_1 Z_2 H$, where $H = X_2 Z_1^2 - X_1 Z_2^2$. If $Z_3 = 0$, we have three cases, (1) $H = 0$, (2) $Z_1 = 0$, or (3) $Z_2 = 0$. By a similar

calculation, we obtain the conditions of the exceptional points which are same as the projective case.

Thus, we have the following theorem.

Theorem 1. *The standard addition formula using the projective (or Jacobian) coordinate for computing $P_1 + P_2$ returns the zero Z-coordinate if and only if one of the following condition satisfies: (1)$P_1 + P_2 = \mathcal{O}$, (2)$P_1 = \mathcal{O}$, (3)$P_2 = \mathcal{O}$ for ECADD(P_1, P_2), or P has order 2 for ECDBL(P).*

3.3 Exceptional Procedure Attack against Standard Formula

We explain the exceptional procedure attack based on the exceptional procedure.

For the sake of simplicity, we assume that the scalar multiplication is computed by the binary method in section 2.3. The scalar multiplication produces the sequence $a_0Q, a_1Q, a_2Q, \ldots, a_nQ$ for the given base point Q, which are generated by ECDBL and ECADD. a_0 is always 2 because of $a_0Q = \mathrm{ECDBL}(Q)$. Then $a_1 = 3$ holds if and only if the second most significant bit $d[n-2]$ is one. If the curve has the point Q with order 3, we can break the second significant bit $d[n-2]$ because of the error $3Q = \mathcal{O}$. Generally the information $a_i (i > 3)$ are able to provide the lower bits $d[n-3], d[n-4]$, and so on.

From Theorem 1, in order to cause the error in the sequence, the attacker has to find the point Q that satisfies one of the following condition (I) ECADD($Q, d_1 * Q$) $= \mathcal{O}$ for some integers d_1, (II) $d_2 * Q = \mathcal{O}$ for some integers d_2, or (III) ECDBL (Q) $= \mathcal{O}$. These cases are equivalent to the problems that find the $(d_1 + 1)$-th division points, d_2-th division points, and the 2-nd division points, respectively. The a-th division points are defined by the points Q that satisfies $a * Q = \mathcal{O}$. It is well-known that the a-th division points exist over the elliptic curve, if and only if $\#E$ is a multiple of a, where $\#E$ is the order of the elliptic curve E [Sil86]. The points with small order can be efficiently generated by the division polynomial. If the order of the curve is divisible by small integers, the curve involves the exceptional points.

The elliptic curves over prime fields recommended in the several standards has prime order [ANSI, IEEE, SEC]. In these cases there are no non-trivial division points and the exceptional point attack against the standard addition formula is not feasible. However, in the next section we show the exceptional procedure attack over the standard curves is effective against a non-standard addition formula from [BrJ02].

3.4 Relation to Other Attacks

Here we examine differences of our attack to other similar attacks.

The Fault Attack (FA) or the Differential Fault Attack (DFA) [BDL97], [BMM00], which is sophisticated from FA, are very similar to our attack. The attack model and the aim are almost same. FA/DFA use special points which are not on the curve. We show a simple example. The attacker changes the curve E and the base point $P \in E$ to E^* and $Q^* \in E^*$ where the

order of E^* is smooth. The result of the scalar multiplication $d*Q^*$ is contained in curve E^*. Thus the attacker can easily recover the secret scalar d by the Pohlig-Hellman algorithm. However, these points are easily detected by checking whether the base point satisfies the curve equation of E or not. On the other hand, our attack uses points on the original curve. The checking process cannot reject the manipulated points. Another difference is the means of the attack. FA/DFA enforce bit errors from outside of the device, while our attack is able to be achieved by a chosen ciphertext attack and we don't need such physical tools.

Other similar attack is the Subgroup Attack (SA) [LMQSV98], which uses a special point on the curve whose order is small. For example, if we use the base point P with the small order $h > 1$, the possible values of the scalar multiplication $d * P$ are at most h. If the attacker can change the base point of the Diffie-Hellman protocol to the smooth order point, then the attack can guess the shared key by the brute-force attack with the size h. SA succeeds only when the order of the curve is divisible by small integer and, if the order is prime, SA has no effectiveness. When we use the standard addition formula, our attack is successful only when the curve has points of order 2. From this point, our attack seems weaker than SA. However, as we will discuss in the next section, our attack is successful for Brier-Joye's addition formula even if the order is prime, while SA has no effectiveness on this curve. Thus our attack is different property from that of SA.

4 Brier-Joye's Addition Formula

In this section, we investigate the security of the non-standard addition formula proposed by Brier and Joye [BrJ02]. The addition formula is designed in order to prevent side channel attacks [KJJ99]. It can computes both ECADD and ECDBL using only one formula. We do not have to switch pairs (λ, μ) of the addition formula depending on inputs. However the addition formula has non-standard exceptional points that have not appeared in the standard addition formula. We analyze these non-standard exceptional points and apply these points to the exceptional procedure attack described in the previous section.

4.1 Brier-Joye's Addition Formula

Let $y(P)$ denote the y-coordinate value of a point P.

Proposition 1 (Indistinguishable Addition Formula, [BrJ02]). *Let E be an elliptic curve over a finite field $\mathbb{F}_p$ ($p > 3$ a prime) defined by $y^2 = x^3 + a\,x + b$ and let $P_1 = (x_1, y_1)$ $P_2 = (x_2, y_2)$ be points on the curve with $y(P_1) \neq y(-P_2)$. Then (λ, μ) in the addition formula is given by*

$$(\lambda, \mu) = \left(\frac{x_1^2 + x_1 x_2 + x_2^2 + a}{y_1 + y_2}, y_1 - \lambda x_1 \right).$$

Brier-Joye also proposed an efficient algorithm to compute $P_1 + P_2$ in the projective coordinate system as follows:

Proposition 2 ([BrJ02]). *Let E be an elliptic curve over a finite field $\mathbb{F}_p$ ($p > 5$ a prime) defined by $Y^2Z = X^3 + a\ XZ^2 + bZ^3$ (the projective coordinate system) and let $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ be points on the curve. Then, $P_3 = (X_3 : Y_3 : Z_3) = P_1 + P_2$ is given by*

$$X_3 = 2FW,\ Y_3 = R(G - 2W) - L^2,\ Z_3 = 2F^3, \tag{2}$$

where $U_1 = X_1Z_2$, $U_2 = X_2Z_1$, $T = U_1 + U_2$, $R = T^2 - U_1U_2 + a\ Z^2$, $M = Y_1Z_2 + Y_2Z_1$, $F = Z_1Z_2M$, $L = MF$, $G = TL$ and $W = R^2 - G$.

4.2 Exceptional Procedure in Brier-Joye's Formula

Let $P_1 = (X_1 : Y_1 : Z_1)$, $P_2 = (X_2 : Y_2 : Z_2)$, $P_3 = (X_3 : Y_3 : Z_3) = P_1 + P_2$ be points on the curve represented in the projective coordinate. If $Z_3 = 0$, from (2), we have three cases, (1) $Y_1Z_2 + Y_2Z_1 = 0$, (2) $Z_1 = 0$, or (3) $Z_2 = 0$. The latter two cases are reduced to trivial conditions $P_1 = \mathcal{O}$ or $P_2 = \mathcal{O}$. However the first condition is worth to investigate. The condition implies $y_1 + y_2 = 0$. If $P_1 + P_2 = \mathcal{O}$, we have $y_1 + y_2 = 0$, but this is not interesting. Conversely, even if $y_1 + y_2 = 0$, $P_1 + P_2$ does not always equal to $\mathcal{O}$. That is, we can pick up points $P_1,\ P_2$ such that $x_1 \neq x_2,\ y_1 + y_2 = 0$. Once such "exceptional points" are added in the scalar multiplication, we have $Z_d = 0$ and the conversion from the projective to the affine fails. In this case, we cannot obtain the correct result of $d * P = (x_d, y_d)$ and we can observe that an error has occurred in the scalar multiplication.

4.3 Finding Collision Points

Next, we discuss the criteria $y_1 + y_2 = 0$, which are exceptional cases of the Brier-Joye's addition formula. We call two points $P_1 = (x_1, y_1),\ P_2 = (x_2, y_2)$ satisfy the *DZ condition* if

$$x_1 \neq x_2,\ y_1 + y_2 = 0$$

holds, and in this case, we call P_1, P_2 as a *collision pair*. The necessary condition for the DZ condition is $x_1^3 + a\ x_1 + b = x_2^3 + a\ x_2 + b$, namely $x_1^2 + x_1x_2 + x_2^2 + a\ = 0$. From the condition we can generate a collision pair P_1, P_2, which satisfies the DZ condition. A point P is called the *m-th self-collision point* if P and $m * P$ is the collision pair.

We explain how to find a collision pair (P_1, P_2) in the following. For a given elliptic curve $E : y^2 = x^3 + ax + b$ and a base point $P_1 = (x_1, y_1)$ on the curve, determining whether P_1 has collision points or not is easy. For simplicity, we assume the order of the elliptic cure E is prime. If (P_1, P_2) is a collision pair, an intuitive relation of P_1 and P_2 is in Fig. 1. So, P_1 has collision points if the equation $x^2 + x_1x + (x_1^2 + a) = 0$ has roots in $\mathbb{F}_p$ and this evaluation

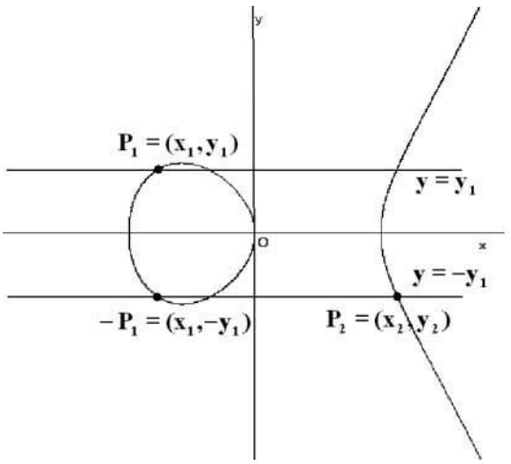


Fig. 1. A geometric relation of collision points

is done quite easily. However, we need a relation between P_1 and P_2 in the attack, namely we have to solve the discrete logarithm $P_2 = u * P_1$ on the curve (the *Collision-ECDLP*). This problem might be easier than the general discrete logarithm problem over elliptic curves because we have the constrained condition $x_1 \neq x_2$ and $y_1 + y_2 = 0$. However there is no evidence of the difference between these problems and this is an open problem.

Thus we have to change the approach. Assume we have an elliptic curve E and an integer m. The next approach is to find a point P_1 such that $(P_1, m * P_1)$ is a collision pair. Such P_1 satisfies a certain equation – the *self-collision polynomial*, which will be defined in the next section – and finding P_1 is equivalent to solve this equation. Roughly speaking, computing the m-th self-collision polynomial is not easier than computing the m-th division polynomial at the moment. However, computing the m-th self-collision points is feasible for small m, which are enough for our attack.

4.4 Self-Collision Polynomial

We discuss how to find the m-th self-collision points for a randomly chosen curve. We denote the m-th division polynomial as $\psi_m = \psi_m(x, y)$. If a point $P = (x, y)$ is in the m-torsion group, namely $m * P = \mathcal{O}$, then (x, y) satisfies $\psi_m(x, y) = 0$. Let denote $P = (x, y)$ and $m * P = (x_m, y_m)$. Then, x_m and y_m are written as in the following by the division polynomials [BSS99]:

$$(x_m, y_m) = \left(x - \frac{\psi_{m-1}\psi_{m+1}}{\psi_m^2}, \frac{\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2}{4y\psi_m^3} \right). \tag{3}$$

If P and $m * P$ is a collision pair, we have $y + y_m = 0$ and so

$$F_m(x, y) = 4y^2\psi_m^3 + \psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2 = 0. \tag{4}$$

On the other hand, because of $y^2 = y_m^2$ and $x - x_m \neq 0$, we have

$$G_m(x, y) = (3x^2 + a)\psi_m^4 - 3x\psi_m^2\psi_{m-1}\psi_{m+1} + \psi_{m-1}^2\psi_{m+1}^2 = 0. \tag{5}$$

Here the two equations $F_m(x, y)$ and $G_m(x, y)$ have a common polynomial divisor $f_m(x, y)$. Small examples of $f_m(x)$ are in the appendix. A concrete relation between $F_m(x, y)$ and $G_m(x, y)$ is given by the following proposition. The proof is described in the appendix.

Proposition 3. *Let m be an integer $m \geq 2$. Then,*

1. $F_m(x, y) = 4yf_m(x, y)\psi_{m+1}(x, y)$,
2. $G_m(x, y) = f_m(x, y)f_{m+1}(x, y)$.
3. $f_m(x, y) = f_m(x)$, *i.e.* $f_m \in \mathbb{Z}[x]$
4. $f_m(x) = (m^2 - m + 1)x^{m^2-m} +$ *lower terms of* x

We call the polynomial $f_m(x)$ as the *m-th self-collision polynomial*. As in the above discussion, if a point $P = (x, y)$ is the m-th self-collision point, $x = x(P)$ should satisfy $f_m(x) = 0$. However all roots of $f_m(x) = 0$ does not lead to the points on the curve. So what we want is roots of $f_m(x) = 0$ such that $x^3 + ax + b$ is quadratic residue. Thus we have the following Theorem:

Theorem 2. *Let $P = (x, y)$ be a point on an elliptic curve. Then, $f_m(x) = 0$ iff P is the m-th self-collision point.*

Corollary 1. *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve. Then, $f_m(x) = 0$ and x^3+ax+b is square iff E has the m-th self-collision points whose x-coordinate value is x.*

We made an experiment of finding the m-the self-collision points for small m ($2 \leq m \leq 9$) using the polynomial $f_m(x)$. We used several standard elliptic curves in the draft of SECG [SEC]. Then we have found several m-th self-collision points. Therefore our proposed attack is feasible for several standard curves with the Brier-Joye's addition formula. These results are summarized in the appendix.

4.5 Attack to the ElGamal-Type Encryption

We shortly explain the exception point attack against the ElGamal-type encryption. The attacker chooses a k-th self-collision point Q on the underlying curve. The point Q is sent to the decryption primitive that computes the scalar multiplication $d * P$ using the secret key d. If the attacker receives the error from the decryption oracle, he/she knows the scalar multiplication has calculated the addition $Q + k * Q$.

We assume that the scalar multiplication is computed by the binary method in section 2. If the attacker wants to guess the 2-nd most significant bit, the

attacker asks the 3-rd self collision point to the decryption oracle, which is computed during the scalar multiplication if and only if the second most significant bit is one. We can recursively apply this process to lower bits.

Note that the k-th self-collision point Q is not a division point such that $Q + k * Q = (k+1) * Q = \mathcal{O}$. Therefore the attack is feasible for the curves with prime order, namely standard curves [ANSI, IEEE, SEC].

5 Concluding Remarks

This paper studied the exceptional procedure attack that uses the exceptional procedure of the addition formula. We show the attack is effective against the addition formula proposed by Brier-Joye. Partial bits of the secret key can be revealed by our proposed attack. We demonstrated the feasibility of our attack against the recommended curves in the international standards [ANSI, IEEE, SEC] and found enough curves for which our attack works. However, the attack discussed in Section 4.5 is restricted to the ElGamal-type systems, in particular it is not relevant to ECDSA because the base point of ECDSA is usually fixed as the system parameter. An application to other cryptosystems will be our future work.

When a new addition formula is designed, the designers should be careful for the exceptional procedure attack. Even though the new formula is secure against previously known attacks, it might be insecure against the exceptional procedure attack or similar attacks based on the exceptional procedures in the formula. This attack can be essentially extended to the attacks against hyperelliptic curve cryptosystems. The security analysis of the attack strongly depends on their explicit formulas.

Acknowledgments

We would like to thank Marc Joye and anonymous referees for their valuable comments.

References

[ANSI] ANSI X9.62, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), draft, 1998. 225, 230, 235

[BMM00] I. Biehl, B. Meyer, and V. Müller, "Differential Fault Attacks on Elliptic Curve Cryptosystems", *CRYPTO 2000*, LNCS 1880, pp.131-146, Springer-Verlag, 2000. 225, 230

[BiJ02] O. Billet and M. Joye, "The Jacobi Model of an Elliptic Curve and Side-Channel Analysis", Cryptology ePrint Archive, Report 2002/125, 2002. 225

[BDL97] D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults", *Eurocrypt'97*, LNCS 1233, pp.37-51, Springer-Verlag, 1997. 225, 230

[BrJ02] E. Brier and M. Joye, "Weierstraß Elliptic Curves and Side-Channel Attacks", *PKC 2002*, LNCS 2274, pp.335-345, Springer-Verlag, 2002. 225, 230, 231, 232

[BSS99] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, 1999. 233

[CMO98] H.Cohen, A.Miyaji and T.Ono, "Efficient Elliptic Curve Exponentiation using Mixed Coordinates", *Asiacrypt'98*, LNCS 1514, Springer-Verlag, pp.51-65, 1998. 227

[IEEE] IEEE P1363, Standard Specifications for Public-Key Cryptography, 2000. Available from `http://groupe.ieee.org/groups/1363/` 225, 230, 235

[IT02] T. Izu and T. Takagi, "On the Security of Brier-Joye's Addition Formula for Weierstrass-form Elliptic Curves", Technical Report, No. TI-3/02, Technische Universität Darmstadt, 2002.

[JQ01] M. Joye and J. Quisqiater, "Hessian Elliptic Curves and Side-Channel Attacks", *CHES 2001*, LNCS 2162, pp.412-420, Springer-Verlag, 2001. 225

[KJJ99] C. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis", *Crypto'99*, LNCS 1666, pp.388-397, Springer-Verlag, 1999. 225, 231

[LMQSV98] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An Efficient Protocol for Authenticated Key Agreement", Technical report CORR 98-05, University of Waterloo, 1998. 225, 231

[LS01] P. Liardet and N. Smart, "Preventing SPA/DPA in ECC System using the Jacobi Form", *CHES 2001*, LNCS 2162, pp.401-411, Springer-Verlag, 2001. 225

[OKS00] K. Okeya, H. Kurumatani, and K. Sakurai, "Elliptic Curves with the Montgomery Form and their cryptographic Applications", *PKC 2000*, LNCS 1751, pp.446-465, Springer-Verlag, 2000. 225

[SEC] Standards for Efficient Cryptography Group (SECG), Specification of Standards for Efficient Cryptography. Available from `http://www.secg.org` 225, 230, 234, 235, 236

[Sil86] J. Silverman, *The Arithmetic of Elliptic Curves*, GMT 106, Springer-Verlag, 1986. 230

[Sma01] N. Smart, "The Hessian Form of an Elliptic Curve", *CHES 2001*, LNCS 2162, pp.118-125, Springer-Verlag, 2001. 225

A Numerical Examples

In this appendix, we show numerical examples of polynomial $f_m(x)$ and the m-th self-collision points on standardized curves over a prime field in [SEC].

A.1 Self-Collision Points

Table 1 shows the number of the m-th self-collision points ($2 \leq m \leq 9$) on the elliptic curves standardized in [SEC].

Table 1. The number of the m-th self-collision points

m	2	3	4	5	6	7	8	9
secp112r1	2	-	-	2	2	-	-	2
secp112r2	-	-	-	2	-	-	2	4
secp128r1	-	2	4	-	-	2	4	-
secp128r2	2	-	-	-	2	-	2	2
secp160k1	-	-	-	-	-	-	-	-
secp160r1	-	-	-	-	-	-	-	2
secp160r2	-	-	2	-	-	-	-	-
secp192k1	-	-	-	-	-	-	-	-
secp192r1	2	-	-	2	2	-	4	-
secp224k1	-	-	-	-	-	-	-	-
secp224r1	-	2	4	-	-	2	2	2
secp256k1	-	-	-	-	-	-	-	-
secp256r1	-	-	-	-	4	2	-	-
secp384r1	2	-	-	2	2	2	-	-
secp521r1	4	-	2	-	-	-	-	-

In the following, the numerical data of self-collision points on the standardized curve `secp128r1` are listed. All data are described in hexadecimal.

$$p = \texttt{0xfffffffdffffffffffffffffffffffff}$$
$$a = \texttt{0xfffffffdfffffffffffffffffffffffc}$$
$$b = \texttt{0xe87579c11079f43dd824993c2cee5ed3}$$

The 3rd self-collision points (2 points)

(`0xa2b4652401379e1e3ff1f915e64ca2c8`, `0xea7c93c0989bb3d2d4611a81df3032b`)
(`0xa2b4652401379e1e3ff1f915e64ca2c8`, `0xf15836c1f67644c2d2b9ee57e20cfcd4`)

The 4-th self-collision points (4 points)

(`0xfc34bdc223c2601307ad0b8b21e1c8be`, `0xc4e0ed0ac0db88cf58ee1806bc2621e`)
(`0xfc34bdc223c2601307ad0b8b21e1c8be`, `0xf3b1f12d53f247730a711e7f943d9de1`)
(`0x28a53b1ca02fdb170f2330225b05cab6`, `0xdfed12d13ceba387b3695ef16010f0f7`)
(`0x28a53b1ca02fdb170f2330225b05cab6`, `0x2012ed2cc3145c784c96a10e9fef0f08`)

The 7-th self-collision points (2 points)

(`0x3e420615cdb89fc6b57989b0661d13a8`, `0x23dc8dd9995872ba72a1cbccfbffe4ff`)
(`0x3e420615cdb89fc6b57989b0661d13a8`, `0xdc23722466a78d458d5e343304001b00`)

The 8-th self-collision points (4 points)

(`0x95f75d5e09789632c30aa23aadebd9f7`, `0xbe80ab797a9e63f4a687f081f670e439`)
(`0x95f75d5e09789632c30aa23aadebd9f7`, `0x417f548485619c0b59780f7e098f1bc6`)
(`0x9a8034c28924315a96fc0a0c4f69c358`, `0x9b3a46c0fcce148116e4be42bff777c9`)
(`0x9a8034c28924315a96fc0a0c4f69c358`, `0x64c5b93d0331eb7ee91b41bd40088836`)

A.2 Self-Collision Polynomial $f_m(x)$

Here are small examples of $f_m(x)$. The definition of $f_m(x)$ is in section 4.3.

$$
\begin{aligned}
f_2(x) &= 3x^2 + a \\
f_3(x) &= 7x^6 + 11a\ x^4 - 4bx^3 + 13a^2x^2 + 20a\ bx + a^3 + 16b^2 \\
f_4(x) &= 13x^{12} + 70a\ x^{10} + 52bx^9 + 231a^2x^8 + 912a\ bx^7 + (100a^3 + 1536b^2)x^6 \\
&\quad +408a^2bx^5 + (43a^4 + 1776a\ b^2)x^4 + (-176a^3b + 1024b^3)x^3 \\
&\quad +(54a^5 + 96a^2b^2)x^2 + (84a^4b + 448a\ b^3)x + a^6 + 48a^3b^2 + 256b^4 \\
f_5(x) &= 21x^{20} + 298a\ x^{18} + 828bx^{17} + 1917a^2x^{16} + 16224a\ bx^{15} \\
&\quad +(-360a^3 + 43920b^2)x^{14} + 3024a^2bx^{13} + (938a^4 + 88368a\ b^2)x^{12} \\
&\quad +(-31200a^3b + 42432b^3)x^{11} + (11484a^5 + 42768a^2b^2)x^{10} \\
&\quad +(-600a^4b + 113600a\ b^3)x^9 + (13794a^6 + 26928a^3b^2 + 101376b^4)x^8 \\
&\quad +(45216a^5b + 127872a^2b^3)x^7 + (4312a^7 + 104496a^4b^2 + 252672a\ b^4)x^6 \\
&\quad +(16464a^6b + 169344a^3b^3 + 129024b^5)x^5 + (225a^8 + 38160a^5b^2 + 276480a^2b^4)x^4 \\
&\quad +(-1056a^7b + 28352a^4b^3 + 254976a\ b^5)x^3 \\
&\quad +(138a^9 - 720a^6b^2 + 768a^3b^4 + 86016b^6)x^2 + (252a^8b + 1728a^5b^3)x \\
&\quad +a^{10} + 144a^7b^2 + 1536a^4b^4 + 4096a\ b^6 \\
f_6(x) &= 31x^{30} + 967a\ x^{28} + 5332bx^{27} + 10431a^2x^{26} + 162252a\ bx^{25} \\
&\quad +(-37737a^3 + 651744b^2)x^{24} - 233640a^2bx^{23} + (-2373a^4 + 1471536a\ b^2)x^{22} \\
&\quad +(-1775928a^3b - 458304b^3)x^{21} + (755427a^5 + 382896a^2b^2)x^{20} \\
&\quad +(-119844a^4b + 596928a\ b^3)x^{19} + (2161515a^6 + 6446544a^3b^2 + 7594752b^4)x^{18} \\
&\quad +(9080100a^5b + 22216320a^2b^3)x^{17} + (2480643a^7 + 39949488a^4b^2 + 69276672a\ b^4)x^{16} \\
&\quad +(13109904a^6b + 106820352a^3b^3 + 49167360b^5)x^{15} \\
&\quad +(1514205a^8 + 55841760a^5b^2 + 272943360a^2b^4)x^{14} \\
&\quad +(6809520a^7b + 124271232a^4b^3 + 347083776a\ b^5)x^{13} \\
&\quad +(705045a^9 + 34703328a^6b^2 + 295451904a^3b^4 + 158822400b^6)x^{12} \\
&\quad +(482124a^8b + 62178432a^5b^3 + 527431680a^2b^5)x^{11} \\
&\quad +(491997a^{10} + 7532448a^7b^2 + 68961024a^4b^4 + 461328384a\ b^6)x^{10} \\
&\quad +(360276a^9b + 25187328a^6b^3 + 141441024a^3b^5 + 136445952b^7)x^9 \\
&\quad +(273573a^{11} - 1545408a^8b^2 + 15061248a^5b^4 + 225533952a^2b^6)x^8 \\
&\quad +(1294488a^{10}b + 827136a^7b^3 - 30200832a^4b^5 + 132857856a\ b^7)x^7 \\
&\quad +(34569a^{12} + 2980080a^9b^2 + 11748096a^6b^4 - 40587264a^3b^6 + 20643840b^8)x^6 \\
&\quad +(190728a^{11}b + 4920768a^8b^3 + 19031040a^5b^5 - 35979264a^2b^7)x^5 \\
&\quad +(a^{13} + 486768a^{10}b^2 + 7898880a^7b^4 + 26443776a^4b^6 - 31260672a\ b^8)x^4 \\
&\quad +(-5756a^{12}b + 511424a^9b^3 + 8380416a^6b^5 + 29884416a^3b^7 - 10747904b^9)x^3 \\
&\quad +(313a^{14} - 3696a^{11}b^2 + 265728a^8b^4 + 4767744a^5b^6 + 18284544a^2b^8)x^2 \\
&\quad +(572a^{13}b + 7040a^{10}b^3 + 135168a^7b^5 + 1622016a^4b^7 + 5767168a\ b^9)x \\
&\quad +a^{15} + 304a^{12}b^2 + 5888a^9b^4 + 61440a^6b^6 + 393216a^3b^8 + 1048576b^{10}
\end{aligned}
$$

B Proof of Proposition 3

Proposition 3. *Let m be an integer $m \geq 2$. Then,*

1. $F_m(x,y) = 4yf_m(x,y)\psi_{m+1}(x,y)$,
2. $G_m(x,y) = f_m(x,y)f_{m+1}(x,y)$.
3. $f_m(x,y) = f_m(x)$, *i.e.* $f_m \in \mathbb{Z}[x]$
4. $f_m(x) = (m^2 - m + 1)x^{m^2-m}+$ *lower terms of* x

Proof. The division polynomial ψ_m is a polynomial in $\mathbb{Z}[x]$ if m is odd, and $\psi_m/(2y)$ is a polynomial in $\mathbb{Z}[x]$ if m is even. 1. If a point $P = (x,y)$ satisfies $\psi_{m+1} = 0$, then P is the m-th self-collision. So, we have $\psi_{m+1}(x,y)|F_m(x,y)$. If m is odd, we have $4y^2|\psi_{m+2}\psi_{m-1}^2$, $4y^2|\psi_{m-2}\psi_{m+1}^2$ and $4y|F_m(x,y)$. It is the same for even m. 2. If $P = (x,y)$ is the $(m+1)$-th self-collision, then, $-P = (x,-y)$ is the m-th self-collision. So we have $f_{m+1}(x,y)|G_m(x,y)$. 3. If m is odd, $F_m(x,y) \in \mathbb{Z}[x]$. On the other hand, ψ_{m+1} can be factored into the form $2yg(x)$. So $f_m = F_m/(8y^2g(x)) \in \mathbb{Z}[x]$. It's the same for even m. 4. We know $\psi_m(x,y) = mx^{(m^2-1)/2}+$ lower term of x, where we weight x as 1 and y as $3/2$.