Technical Report, Technische Universität Darmstadt, No.TI-3/02, 27.02.2002.
URL: http://www.informatik.tu-darmstadt.de/TI/Veroeffentlichung/TR/

On the Security of Brier-Joye's Addition Formula for Weierstrass-form Elliptic Curves

Tetsuya Izu* Tsuyoshi Takagi**

* FUJITSU LABORATORIES Ltd.,
4-1-1, Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588, Japan
izu@flab.fujitsu.co.jp

** Technische Universität Darmstadt, Fachbereich Informatik,
Alexanderstr.10, D-64283 Darmstadt, Germany
ttakagi@cdc.informatik.tu-darmstadt.de

Abstract

Elliptic curve based cryptosystems (ECC) are thought to be suitable for implementing on low-power devices such as smart cards because of the small key-length. The side channel attacks against these devices may be successful if the implementation is naive or careless. Recently, Brier and Joye proposed the indistinguishable addition formula for Weierstrass-form elliptic curves [BJ02] in order to resist the side channel attacks. This paper studies the property of the formula and proposes a chosen ciphertext attack to ElGamal-type cryptosystems with their addition formula. We demonstrate several standard curves recommended in [SEC] can be attacked by our proposed attack. We also extend the formula to Montgomery-form elliptic curves but conclude that it has the same problem as Weierstrass-form case.

Keywords: elliptic curve based cryptosystem (ECC), scalar multiplication, side channel attack, indistinguishable addition formula

1 Introduction

As the importance of security increases, the authentication becomes one of the vital technique in this area. When we are to authenticate on the low-power devices such as smart cards and mobile phones, we need an efficient cryptosystem for these devices. Elliptic curve based cryptosystems are thought to be suitable for implementing on such devices because of the small key-length. However, if the implementation is naive or careless, the side channel attacks (SCA) may allow an adversary to reveal the secret key in the cryptographic device by observing the side channel information such as the computing time and the power consumption [Koc96, KJJ99]. The adversary does not have to break the device physically to obtain the secret key. The simple power analysis (SPA) only uses a single

observed information, while the differential power analysis (DPA) uses a lot of observed information together with statistic tools.

There are three approaches to resist the SPA. The first one is the add-and-double-always method which masks the scalar dependency by inserting dummy operations. The Coron's algorithm [Cor99] is an example. The second one uses the Montgomery's ladder [Mon87], which essentially resists the SPA [OKS00, IT02, BJ02, FGKS02]. The third one establishes the indistinguishable addition and doubling in the scalar multiplication [CJ01]. For a binary field case, Bellezza proposed an indistinguishable adding and a doubling for Weierstrass-form curves by inserting dummy operations [Bel01]. For a prime field case, only Hesse- and Jacobi-form elliptic curves have achieved the indistinguishability [LS01, JQ01]. Because of the specialty of these curves, they are not compatible to the standardized curves in [ANSI, IEEE, SEC]. Recently, Brier-Joye proposed an indistinguishable addition and doubling for Weierstrass-form elliptic curves for both binary and prime fields [BJ02]. Their formulation is very simple and looks natural. However the security consideration was not enough.

Contribution of this paper

The purpose of this paper is to study the property of the Brier-Joye's addition formula (BJ's formula for short). We show some problems in the formula and propose a chosen ciphertext attack with the help of the side channel information against the BJ's formula. An attacker generates an appropriate point X on the elliptic curve E and observes the side channel information of the decryption uX for the target secret key u. By the proposed attack, we can obtain the partial bits of the secret key u. The leaked bits depend on the addition chain used in the scalar multiplication.

We call two points $P_1,\ P_2$ on an elliptic curve satisfy the DZ condition if $x(P_1) \neq x(P_2)$ and $y(P_1) = y(-P_2)$ holds. The points that satisfy the DZ condition are exceptional points of the BJ's formula. They are different from the exceptional points of the standard addition formula, namely $x(P_1) = x(P_2)$ and $y(P_1) = y(\pm P_2)$. If we compute $P_1 + P_2$ that satisfy the DZ condition, BJ's formula causes an error in affine coordinate because it cannot compute the inverse of the denominator. In the projective/Jacobian coordinate it returns the point whose Z-coordinate is 0. In the scalar multiplication, once the Z-coordinate is 0, the following values of the Z-coordinate in the addition chain also become 0 and the scalar multiplication causes an error when the final result is recovered to the affine coordinate.

Here we assume that the scalar multiplication is computed by a standard binary multiplication. Let $X \in E$ be a point, where $2X$ and X satisfy the DZ condition. We utilize the DZ condition for our proposed attack. The proposed attack is a chosen ciphertext attack and it asks the decryption oracle to computes uX for a chosen point X with unknown secret u. If the scalar multiplication causes an error, we know the second most significant bit is 1, otherwise 0. In general, if we generate a point $X \in E$, where dX and X satisfy the DZ condition for some integer d, we can guess other bits. Note that the points which satisfy the DZ condition are generally not torsion points X such that $dX = \mathcal{O}$ for some integer d and thus our proposed attack is different from the small subgroup attack [LMQSV98]. Moreover, our attack is different from the fault based attacks [BDL97] [BMM00].

We discuss the success probability of our attack for a general elliptic curve. Moreover, we demonstrate the feasibility of our proposed attack against the recommended curves by the international standards [ANSI, IEEE, SEC]. We check there are enough curves for which our attack works.

The attack proposed in this paper 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. In the case of randomization of the multiplier (Coron's 1st countermeasure against the DPA [Cor99]) the attack is no longer successful.

Finally, we apply BJ's formulation to the Montgomery-form elliptic curves. As an indistinguishable adding/doubling can be established, the same problem for the original formulation remains, too.

2 Addition Formula

In this section we review the standard addition formula and Brier-Joye's addition formula (BJ's formula, for short). Let $\mathbb{F}_q$ be a finite field with $q = p^m$ elements, where p is a prime. Let E be an elliptic curve over $\mathbb{F}_q$ defined by Weierstrass-form equation

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6.$$

A set of points on the curve, including the point of infinity $\mathcal{O}$, is denoted by $E(\mathbb{F}_q)$. This set has an additive group structure and the addition rule is given by the addition formula.

2.1 Standard Formula

A standard addition formula consists of an addition $P_1 + P_2$ ($P_1 \neq \pm P_2$) and a doubling $2P_1$, where P_1, P_2 are points on E. The formulas for computing the addition and the doubling are different. If $P_1 = -P_2$, we define $P_3 = P_1 + P_2 = \mathcal{O}$. If not, for two points $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ on the curve, we define $P_3 = P_1 + P_2 = (x_3, y_3)$ by the following:

$$\begin{aligned} x_3 &= \lambda^2 + a_1\lambda - a_2 - x_1 - x_2, \\ y_3 &= -(\lambda + a_1)x_3 - \mu - a_3, \end{aligned}$$

where

$$(\lambda, \mu) = \begin{cases} (\frac{y_2 - y_1}{x_2 - x_1}, \frac{y_1x_2 - y_2x_1}{x_2 - x_1}) & x_1 \neq x_2 \\ (\frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, \frac{-x_1^3 + a_4x_1 + 2a_6 - a_3y_1}{2y_1 + a_1x_1 + a_3}) & x_1 = x_2, P_2 \neq -P_1 \end{cases}$$

Note that we have two formulas for (λ, μ). The first and the second formula are called ECADD and ECDBL, respectively. The exceptional points of the standard formula are the points $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$, which satisfy $P_1 = \pm P_2$, namely $x_1 = x_2$ and $y_1 = \pm y_2$.

2.2 Brier-Joye's Formula

When we use the standard addition formula, we have to choose one of the pairs (λ, μ) depending on the bit information. Therefore attackers can distinguish which formula have been used in the addition chain by analyzing the side channel information. This is a basic idea of the simple power analysis (SPA). The reason why this attack can be applied is the direct connection between the bit information of the secret key and the choice of an adding/doubling.

Brier and Joye proposed an addition formula, which computes an addition and a doubling with the same formula [BJ02]. We do not have to switch the pair (λ, μ). We describe Brier-Joye's addition formula in the following. $y(P)$ denotes the y-coordinate value of a point P.

Proposition 1 (Indistinguishable Addition Formula, Proposition 1, [BJ02]) *Let E be an elliptic curve over a finite field $\mathbb{F}_q$ defined by $y^2+a_1xy+a_3y = x^3+a_2x^2+a_4x+a_6$ and let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ be points on the curve with $y(P_1) \neq y(-P_2)$. Then (λ, μ) in the standard addition formula is given by*

$$(\lambda, \mu) = \left(\frac{x_1^2 + x_1x_2 + x_2^2 + a_2x_1 + a_2x_2 + a_4 - a_1y_1}{y_1 + y_2 + a_1x_2 + a_3}, y_1 - \lambda x_1 \right).$$

Corollary 2 (Prime Field, Corollary 1, [BJ02]) *Let E be an elliptic curve over a finite field $\mathbb{F}_p$ ($p > 5$ a prime) defined by $y^2 = x^3+ax+b$ 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,*

$$(\lambda, \mu) = \left(\frac{x_1^2 + x_1x_2 + x_2^2 + a}{y_1 + y_2}, y_1 - \lambda x_1 \right).$$

Corollary 3 (Binary Field, Corollary 2, [BJ02]) *Let E be an elliptic curve over a finite field $\mathbb{F}_{2^m}$ defined by $y^2 + xy = x^3 + ax^2 + b$ 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,*

$$(\lambda, \mu) = \left(\frac{x_1^2 + x_1x_2 + x_2^2 + ax_1 + ax_2 + y_1}{y_1 + y_2 + x_2}, y_1 - \lambda x_1 \right).$$

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

Proposition 4 ([BJ02]) *Let E be an elliptic curve over a finite field $\mathbb{F}_p$ ($p > 5$ a prime) defined by $Y^2Z = X^3 + aXZ^2 + bZ^3$ (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*

$$\begin{cases} X_3 &= 2FW, \\ Y_3 &= R(G - 2W) - L^2, \\ Z_3 &= 2F^3, \end{cases} \tag{1}$$

where $U_1 = X_1Z_2$, $U_2 = X_2Z_1$, $S_1 = Y_1Z_2$, $S_2 = Y_2Z_1$, $Z = Z_1Z_2$, $T = U_1 + U_2$, $M = S_1 + S_2$, $R = T^2 - U_1U_2 + aZ^2$, $F = ZM$, $L = MF$, $G = TL$ and $W = R^2 - G$. P_3 can be computed with 18 multiplications in $\mathbb{F}_p$.

2.3 Observation

Here presents an observation for the Brier-Joye's addition formula. For a simplicity, we deal with the Brier-Joye's addition formula only for prime fields in the following. However, the similar discussion can be applied to the rest cases. We consider the case $y(P_1) = y(-P_2)$ in Brier-Joye's formula. This interest comes from Brier-Joye's transformation to obtain their λ because they multiply $y(P_1) - y(-P_2)$ to both denominator and numerator of the original λ.

First observation is trivial but crucial for our attack. If two points P_1 and P_2 satisfy $P_1 + P_2 = \mathcal{O}$, we have $y(P_1) = y(-P_2)$. Conversely, even if $y(P_1) = y(-P_2)$ holds, $P_1 + P_2$ does not always equal to $\mathcal{O}$. That is, when we compute the scalar multiplication dP with Brier-Joye's formula, we may compute $P_1 + P_2$ satisfying $y(P_1) = y(-P_2)$, which is the exceptional case of Brier-Joye's addition formula and causes an error. Of curse, one can handle this exceptional case by using special addition formula or other procedures. However, as the basic idea of the SPA is based on the dependency between the bit information of the scalar and the addition formula to be used, this exceptional case can be easily detected by the SPA.

When we compute the scalar multiplication $uP = (x_u, y_u)$, the projective coordinate system is used in order to avoid computing inversions. Note that the condition $y(P_1) = y(-P_2)$ is equivalent to $Y_1Z_2 + Y_2Z_1 = 0$ in projective coordinate system. If the input for the Brier-Joye's formula satisfies this relation, the Z-coordinate of the output is 0 because of $Z_3 = 2Z_1^3Z_2^3(Z_1Z_2(Y_1Z_2 + Y_2Z_1))^3 = 0$. Once Z-coordinate becomes 0 during the computation of the scalar multiplication, Brier-Joye formula (1) always outputs $Z_3 = 0$ and thus we obtain $Z_u = 0$. At the end of the scalar multiplication, the projective coordinate point $uP = (X_u : Y_u : Z_u)$ is converted to the affine coordinate (x_u, y_u) by computing $x_u = X_u/Z_u$, $y_u = Y_u/Z_u$. Here the conversion to the affine coordinate fails and we cannot obtain the correct result of $uP = (x_u, y_u)$. Moreover, an attacker can detect the failure by the side channel attack and obtain the bit information of the secret key.

2.4 Finding Collision Points

In the proposed attack, collision points (P_1, P_2) have an important role. 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 $x^2 + sx + t = 0$ has roots in $\mathbb{F}_p$, where

$$x^3 + ax + (b - y_1^2) = (x - x_1)(x^2 + sx + t),$$

and this evaluation 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 $P_2 = uP_1$ on the curve. 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.

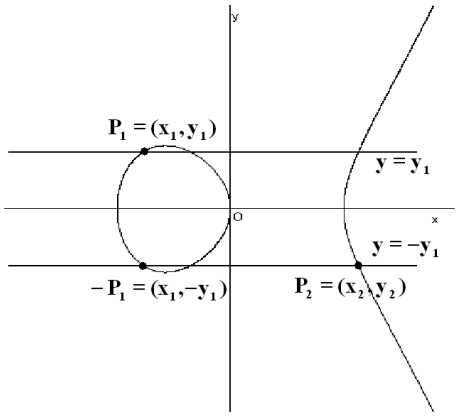

Figure 1: A geometric relation of collision points

Thus we have to change the approach. Assume we have an elliptic curve E and an integer d. The next approach is to find a point P_1 such that (P_1, dP_1) is a collision pair. We call P_1 the d-th self-collision point if P_1 and dP_1 satisfies the DZ-condition. Such P_1 satisfies a certain equation – the self-collision polynomial, which will be defined in Section 4 – and finding P_1 is equivalent to solve this equation. Roughly speaking, computing the d-th self-collision polynomial is as hard as computing the d-th division polynomial. The computation of finding the d-th self-collision points is feasible for small d. They are enough for our attack.

3 Attack Model

In this section we specify our attack model. We consider the chosen ciphertext attack (CCA) [BDPR98]. After the invention of the CCA against PKCS #1 version 1.5 [Ble98], the CCA becomes a standard security criteria for a general purpose usage of public-key cryptosystems. Brier-Joye's addition formula is proposed to prevent from the side channel attack. Therefore we mount the CCA to the side channel attack. An attacker can observe side channel information during the decryption computation for chosen cipher texts.

This type of the chosen ciphertext attacks against the RSA based cryptosystems has been proposed: Schindler reported a timing attack against the RSA with the Chinese remainder theorem in the case of using the Montgomery multiplication [Sch00]. Manger proposed a timing attack against the PKCS #1 version 2.0 under the CCA [Man01]. Novak reported an SPA-based chosen ciphertext attack against the RSA with the Chinese remainder theorem [Nov02].

The target of our attack is the ElGamal cryptosystem over elliptic curves. Our attack utilizes special points of elliptic curves, which cause an error if Brier-Joye's addition formula is used. In the chosen ciphertext attack model, an attacker can select an appropriate ciphertext and ask the decryption oracle to decrypt the ciphertext. Moreover, the attacker can observe the side channel information correlated to the ciphertext. He/She is able to guess the secret key by the side channel information of the chosen points.

3.1 ElGamal Cryptosystem

We describe the ElGamal cryptosystem that we analyze in the following. Denote by T the domain parameter of the ElGamal cryptosystem. If the base field is a prime field, $T = (p, a, b, P, n, h)$, where p is characteristic of the base field, a, b are the coefficients of the definition equation $y^2 = x^3 + ax + b$, P is the base point, n is the order of P, and h is the cofactor of the order of the curve. If the characteristic of the base field is 2, the domain parameter is $T = (m, f(x), a, b, P, n, h)$, where m is the extension degree, $f(x)$ is the definition equation of the base field, a, b are the coefficients of the definition equation $y^2 + xy = x^3 + ax^2 + b$, P is the base point, n is the order of the base point P, and h is the cofactor of the order of the curve.

Let u, Q_U be keys of an user U, where u is the secret key, Q_U is the public key, which satisfy $Q_U = uP$. A message M is a point of the elliptic curve E. The encryption and the decryption of the ElGamal cryptosystem is as follows:

- ENCRYPTION:
  1. Compute $C_1 = rP$ for a random integer $0 \leq r < n$.
  2. Compute $C_2 = M + rQ_U$.
  3. Return the ciphertext $C = (C_1, C_2)$ of the message M.
- DECRYPTION:
  1. Compute $R = uC_1$.
  2. Return the message $M' = C_2 - R$.

3.2 Chosen Ciphertext Attack

We describe the chosen ciphertext attack against the ElGamal cryptosystem. A classical chosen attack can recover the encrypted message by asking a query to the decryption oracle. Our chosen ciphertext attack can break the partial information of the secret key.

The first component of the ciphertext C is a random point of E because $C_1 = rP$ is generated by the random integer r. Even if an attacker changes C_1 to a different point, the decryption oracle cannot distinguish it from the proper points before finishing the decryption. Here the attacker chooses an appropriate $X \in E$ and a random point $C_2 \in E$. Then we assign $C = (X, C_2)$ as a ciphertext. The decryption oracle decrypts the ciphertext based on the secret key u, which consists of two computations $R = uX$ and $M' = C_2 - R$. The attack will guess the secret key based on the side channel information of the computation $R = uX$ for the chosen X.

- Generation of a ciphertext:
  1. Choose an appropriate $X \in E$.
  2. Choose random $C_2 \in E$.

3. Assign $C = (X, C_2)$ as a ciphertext.

- Decryption oracle:
  1. Compute $R = uX$.
  2. Return the message $M' = C_2 - R$.

The ElGamal cryptosystem is extended to provably secure versions in order to prevent from the chosen ciphertext attack [BDPR98]. Most provably secure variants are converted from the original ElGamal cryptosystem with control padding using the random oracle model [KCJLMWY01]. In their decryption process, they first perform the classical decryption $R = uX$ using the secret key u, and then check the control padding. We can thus choose an appropriate point X for our attack. These provably secure variants of the ElGamal cryptosystem can also be broken by our attack in the same manner.

3.3 Side Channel Attack

The side channel attack (SCA) is a serious attack against mobile devices. The SCA allows an adversary to reveal the secret key in the device by observing side channel information such as computing time and power consumption [Koc96, KJJ99]. An adversary does not have to break the devise physically to obtain the secret key. The simple power analysis (SPA) only uses a single observed information, while the differential power analysis (DPA) uses a lot of observed information together with statistic tools.

We explain the side channel attacks that we consider in this paper. The side channel information related to the decryption of uX using secret key u is utilized. For the sake of simplicity, we assume the scalar multiplication uX is computed by the standard binary addition chain. Let $u = u[0]2^0 + u[1]2^1 + ... + u[k-1]2^{k-1}$ be the binary representation of u, where $u[k-1]$ is the most significant bit of u and $u[k-1] = 1$. Then the binary addition chain computes the scalar multiplication uX for given $u[0], u[1], ..., u[k-1]$ and X in the following.

```
INPUT u, X, (u[0],u[1],..,u[k-1])
OUTPUT u*X
 1: Q=X
 2: for i=k-2 down to 0
 3:   Q = 2Q
 4:   if u[i]==1
 5:     Q = Q + X
 6: return Q
```

If we use a standard addition formula for the ECDBL in step 3 and the ECADD in step 5, the bit information can be detected by the SPA [Cor99]. Both ECDBL and ECADD are implemented by arithmetic of the base field, and the ECDBL and ECADD have the different power consumption trace. An side channel attacker can easily distinguish the ECDBL and the ECADD. Then he/she can also know the bit information $u[i]$ for $i = 1, 2, ..., k-1$. Coron proposed an SPA-resistant scheme using dummy operations [Cor99]. This methods is one of standard methods to prevent the side channel attack. The algorithm is as follows:

```
INPUT u, X, (u[0],u[1],..,u[k-1])
OUTPUT u*X
 1: Q[0]=X
 2: for i=k-2 down to 0
 3:   Q[0] = 2Q[0]
 4:   Q[1] = Q[0] + X
 5:   Q[0] = Q[u[i]]
 6: return Q[0]
```

However, we show how to break the bit $u[i]$ of the Coron's algorithm using the exceptional points of the Brier-Joye's addition formula. If the Brier-Joye's addition formula computes the exceptional points, it makes an error in affine coordinate or returns Z-coordinate as 0 in projective/Jacobian coordinate (See section 2.3). If the Z-coordinate is 0, the point cannot be recovered to affine coordinate and the scalar multiplication returns error. Moreover, an attacker can find the exceptional case by the side channel attack, because we can observe the power consumption trace of the base field operations [Cor99].

4 Proposed Attack

This section describes the proposed attack against the scalar multiplication with Brier-Joye's addition formula. For a simplicity, we consider the attack for a prime field case. The attack can be easily extended to a binary field case.

The attack against Brier-Joye's addition formula is based on the fact: If two points P_1 and P_2 satisfying $x(P_1) \neq x(P_2)$ and $y(P_1) = y(-P_2)$ (DZ condition) are added in Brier-Joye's addition formula, the addition formula cannot compute the addition in affine coordinate. If we compute these points in the projective or Jacobian coordinate, the Z-coordinate of the output point becomes 0. If we add a point with zero Z-coordinate, the following Z-coordinate of the addition is also zero. All Z-coordinates of points in the addition chains after the exceptional addition are 0. An attacker can easily distinguish the difference of the power between the addition with zero Z-coordinate and with non-zero Z-coordinate.

We discuss the criteria $y(P_1) = y(-P_2)$ namely $y_1 + y_2 = 0$, which are exceptional cases of Brier-Joye's addition formula. If $P_1 + P_2 = \mathcal{O}$, then $y_1 + y_2 = 0$ holds. However, $P_1 + P_2 \neq \mathcal{O}$ is possible even if $y_1 + y_2 = 0$. 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. We call P_1, P_2 as a *collision pair*. The necessary condition for the DZ condition is

$$\begin{aligned} x_1^3 + ax_1 + b &= x_2^3 + ax_2 + b \\ (x_1 - x_2)(x_1^2 + x_1x_2 + x_2^2 + a) &= 0 \\ x_1^2 + x_1x_2 + x_2^2 + a &= 0. \end{aligned}$$

From the condition we can generate a collision pair P_1, P_2, which satisfies the DZ condition. For a scalar d, a point P is called the *d-th self-collision point* if P and dP is the collision pair.

4.1 Outline of the Attack

We explain the outline of our proposed attack. An attacker chooses a point X which satisfies the DZ condition for X and dX with an integer d. The point X is the d-th self-collision point. The attacker observes the power consumption of the decryption dX.

Here we assume that the value dX is computed using the Coron's method in section 3.3. Note that $2X + X$ is computed and chosen for the next loop if and only if the second most significant bit is 1. If the second most significant bit is 0, we have the DZ condition in the dummy operation $2X + X$. However the output of the dummy operation is not chosen for the next loop, and the final result of the scalar multiplication does not cause error. Therefore the final error for the Coron's algorithm is same as that for the standard binary method. We will consider the security of the standard binary method in the following.

Once an attacker chooses the 2-nd self-collision point X, the attacker can know the second most significant bit. Let u be the k-bit secret key and let $u[i]$ be the i-th bit of u. The proposed attack is constructed as follows:

1. Guess of bit $u[k-2]$:

   (1.1) Generate the 2-nd self-collision point $X \in E$.

   (1.2) Send the ciphertext (X, C_2) to the decryption oracle, where C_2 is a random point of E.

   (1.3) If the scalar multiplication uX causes error, return $u[k-2] = 1$. Otherwise, return $u[k-2] = 0$.

2. Guess of bit $u[k-3]$:

   (2.1) Generate the 6-th self-collision point $X \in E$ if $u[k-2] = 1$. Otherwise, generate the 4-th self-collision point $X \in E$ for $u[k-2] = 0$.

   (2.2) Send the ciphertext (X, C_2) to the decryption oracle, where C_2 is a random point of E.

   (2.3) If the scalar multiplication uX causes error, return $u[k-3] = 1$. Otherwise, return $u[k-3] = 0$.

In order to guess bit $u[i] (i = 1, 2, 3, ..)$ from bits $u[k-2], ..., u[i+1]$, we have to generate the point $X \in E$ such that X and

$$2(\cdots 2(2(2X + u[k-2]X) + u[k-3]X) + ... + u[i+2]X) + u[i+1]X \qquad (2)$$

satisfy the DZ condition. In general if we find a d-th self-collision point $X \in E$, with which the scalar multiplication uX of the decryption oracle causes error, we can break the most significant consecutive bits of u from the integer d.

4.2 Estimation of the Success Probability

We estimate the success probability of our proposed attack in the following. The success probability depends on not only the addition chain but also the parameter of curves.

At first we fix the parameters of curves. When we compute the scalar multiplication uX using the binary method (or Coron's method) in section 3.3, some integers d are not appeared in the loop of the addition chain and we can not use the points dX for our proposed attack. Denote by $SCP(i)$ the existence probability of the i-th self-collision point such that (X, iX) satisfy the DZ condition for a fixed curve. We estimate the probability $SCP(i)$ for $i = 1, 2, ...$ The binary method in section 3.3 have the following probability distributions:

$$SCP(2^k + 2i) = 1/2^{k-1}, \quad SCP(2^k + 2i + 1) = 0$$

for $i = 0, 1, ..., 2^{k-1} - 1$ and $k = 1, 2, ...$ Then we have the following theorem:

Theorem 5 *Assume that the decryption oracle computes $uX \in E$ using the standard binary method for a secret key u. Assume that an elliptic curve E has a $(2^k + 2i)$-th self collision point X for some integers k and $i = 0, 1, ..., 2^{k-1} - 1$. If the secret key u is randomly generated, the k most significant consecutive bits of u can be broken with probability $1/2^k$.*

Proof: The probability, which the k-th most significant bit is one, is $1/2$. The probability, which has the value $(2^k + 2i)X$ in the k-th loop of the binary method, is $1/2^{k-1}$. If our attack succeeds, the k most significant consecutive bits of u is known. Thus the total probability of success is $1/2^k$. ■

For example, when we use an elliptic curve that have $(2^8 + 16)$-th collision points, the 8 most significant consecutive bits can be broken with probability $1/256$ by our proposed attack. In other word, if the elliptic curve is used for 256 different random secret keys, one of the secret keys can be broken by the attack.

If we find a k_0-th self collision point for smaller $k_0 (< k)$, the attack does not have to search all possible values of the most significant bits of u. The attack can recursively detect the most significant bits of the secret key u and the probability of theorem 5 is the worst case estimation.

Next, we estimate the success probability of our attack for a random curve. Let P_E the probability that a random curve E has $(2^k + 2i)$-th self-collision point. Then we have the following corollary.

Corollary 6 *Assume that the decryption oracle computes $uX \in E$ using the standard binary method for a secret key u. Let P_E be the existence probability of $(2^k + 2i)$-th self-collision point for an elliptic curve E. If the secret key u is randomly generated, the k most significant consecutive bits of u can be broken with probability $P_E/2^k$.*

If we estimate this probability P_E, we know the total success probability of our proposed attack. Note that there are no calculation of the addition of $(2^k + 2i + 1)X$ and X in the addition chain of the binary method. Even if an elliptic curve has these self-collision points, we cannot break the partial bits of the secret key by our proposed attack. However, if we use a different addition chain like the right-left binary method or the window based methods, there are different types of self-collision points. Therefore, in the next subsection we deal with the d-th self-collision points with $d = 2, 3, ...$ for a randomly chosen elliptic curve.

4.3 Self-Collision Polynomial

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

$$(x_d, y_d) = \left(x - \frac{\psi_{d-1}\psi_{d+1}}{\psi_d^2}, \frac{\psi_{d+2}\psi_{d-1}^2 - \psi_{d-2}\psi_{d+1}^2}{4y\psi_d^3} \right). \tag{3}$$

From the relationship $y + y_d = 0$ of the d-th self-collision point, we have

$$F_d(x, y) = 4y^2\psi_d^3 + \psi_{d+2}\psi_{d-1}^2 - \psi_{d-2}\psi_{d+1}^2 = 0. \tag{4}$$

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

$$G_d(x, y) = (3x^2 + a)\psi_d^4 - 3x\psi_d^2\psi_{d-1}\psi_{d+1} + \psi_{d-1}^2\psi_{d+1}^2 = 0. \tag{5}$$

Here the two equations $F_d(x, y)$ and $G_d(x, y)$ have a common polynomial divisor $f_d(x, y)$ Small examples of $f_d(x)$ are in the appendix . A concrete relation between $F_d(x, y)$ and $G_d(x, y)$ is given by the following proposition.

Proposition 7 *Let d be an integer $d \geq 2$. Then,*

1. $F_d(x, y) = 4yf_d(x, y)\psi_{d+1}(x, y)$,
2. $G_d(x, y) = f_d(x, y)f_{d+1}(x, y)$.
3. $f_d(x, y) = f_d(x)$, *i.e.* $f_d \in \mathbb{Z}[x]$
4. $f_d(x) = (d^2 - d + 1)x^{d^2-d}$+ *lower terms of* x

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

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

Theorem 8 *Let E be an elliptic curve and P is a point on E. Then, $f_d(x) = 0$ iff $P = (x, y)$ is the d-th self-collision point.*

Corollary 9 *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve. Then, $f_d(x) = 0$ and $x^3 + ax + b$ is square iff E has the d-th self-collision points whose x-coordinate value is x.*

Roughly speaking, computing $f_d(x)$ is as hard as computing $\psi_d(x, y)$. It is unknown there exists an efficient algorithm to compute them. In addition, we have to analyze the mathematical property of $f_d(x)$. They remains open problems at the moment.

We made an experiment of finding the d-the self-collision points for small d ($2 \leq d \leq 9$) using the polynomial $f_d(x)$. We used several standard elliptic curves in the draft of SECG [SEC]. Then we have found several d-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.

5 Extension to Montgomery-form Curves

In this section, we discuss the indistinguishable addition formula for Montgomery-form elliptic curves. Let $p > 5$ be a prime. The Montgomery-form elliptic curve over $\mathbb{F}_{p^m}$ is given by the equation $By^2 = x^3 + Ax^2 + x$ ($(A^2 - 4)B \neq 0$). Montgomery-form elliptic curves are used in some cryptographic primitives [OK-ECDH, OK-ECDSA]. A standard addition formula is given by the following.

Proposition 10 (Addition formula for Montgomery-form) *Let E be an elliptic curve over a finite field $\mathbb{F}_{p^m}$ ($p > 5$ a prime) defined by $By^2 = x^3 + Ax^2 + x$ and let $P_1 = (x_1, y_1)$ $P_2 = (x_2, y_2)$ be points on the curve. Then, $P_3 = P_1 + P_2 = (x_3, y_3)$ is given by*

$$\begin{aligned} x_3 &= B\lambda^2 - A - x_1 - x_2, \\ y_3 &= -\lambda x_1 - \mu \end{aligned}$$

where

$$(\lambda, \mu) = \begin{cases} (\frac{y_2 - y_1}{x_2 - x_1}, \frac{y_1 x_2 - y_2 x_1}{x_2 - x_1}) & x_1 \neq x_2 \\ (\frac{3x_1^2 + 2Ax_1}{2By_1}, \frac{-x_1^3 + x_1}{2By_1}) & x_1 = x_2, P_2 \neq -P_1. \end{cases}$$

By the same transformation of Brier-Joye's, we can easily establish the indistinguishable addition formula for Montgomery-form.

Proposition 11 (Indistinguishable addition formula for Montgomery-form) *Under the same assumption as Proposition 10 and $y_1 + y_2 \neq 0$, for both cases, (λ, μ) is given by*

$$(\lambda, \mu) = \left(\frac{(x_1 + x_2)(x_1 + x_2 + A) - x_1 x_2 + 1}{B(y_1 + y_2)}, y_1 - \lambda x_1 \right).$$

Proof: If $y_1 + y_2 \neq 0$,

$$\begin{aligned} \lambda &= \frac{y_2 - y_1}{x_2 - x_1} \cdot \frac{B(y_2 + y_1)}{B(y_2 + y_1)} = \frac{By_2^2 - By_1^2}{B(x_2 - x_1)(y_2 + y_1)} \\ &= \frac{(x_2 - x_1)[(x_2 + x_1)(x_2 + x_1 + A) - x_2 x_1 + 1]}{B(x_2 - x_1)(y_2 + y_1)} \\ &= \frac{(x_2 + x_1)(x_2 + x_1 + A) - x_2 x_1 + 1}{B(y_2 + y_1)} \end{aligned}$$

■

As we established the indistinguishable addition formula for Montgomery-form, however, it is as insecure as original Brier-Joye's formula for Weierstrass-form because, again, Proposition 4 has the exceptional case $y_1 + y_2 = 0$ and our attack is easily extended to Montgomery-form curves.

6 Concluding Remarks

This paper studied the property of Brier-Joye's addition formula for Weierstrass-form elliptic curves and proposed a chosen ciphertext attack combined with the side channel attack to the ElGamal-type cryptosystems using Brier-Joye's formula. The partial bits of the secret key can be broken by our proposed attack. We demonstrated the feasibility of our attack against the recommended curves in the international standards [ANSI, IEEE, SEC] and we found enough curves for which our attack works.

The proposed attack looks similar to the small subgroup attack [LMQSV98] but they are quite different ones. The small subgroup attack is applicable only when the curve has a small cofactor. Our attack does not need such condition. That is, our attack is applicable even when the cofactor is 1. On the other hand, our attack is different from the fault based attacks [BDL97, BMM00]. The fault based attack is an attack in which an attacker breaks the device physically, while our attack only needs the side channel information and does not need such physical attacks.

Our proposed attack utilizes non-standard exceptional points for the addition formula of elliptic curves. When a new addition formula is designed, the designers should be careful for this type of attacks. Even though the new formula is secure against previously known attacks, it might be insecure against our proposed attack or similar attacks based on the exceptional procedures in the formula.

Acknowledgments. We would like to thank Marc Joye for his valuable comments on the side channel attacks.

References

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

[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.

[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.

[BDPR98] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among Notions of Security for Public-Key Encryption Schemes", *CRYPTO'98*, LNCS 1462, pp.26-45, Springer-Verlag, 1998.

[Bel01] A. Bellezza, "Countermeasures against Side-Channel Attacks for Elliptic Curve Cryptosystems", Cryptology ePrint Archive, 2001/103, 2001. Available from `http://eprint.iacr.org/2001/103/`

[Ble98] D. Bleichenbacher, "A Chosen Ciphertext Attack against Protocols based on RSA Encryption Standard PKCS #1", *CRYPTO'98*, LNCS 1462, pp.1-12, Springer-Verlag, 1998.

[BJ02] E. Brier and M. Joye, "WeierstraßElliptic Curves and Side-Channel Attacks", *PKC2002*, LNCS 2274, pp.335-345, Springer-Verlag, 2002.

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

[Cor99] J. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems", *CHES'99*, LNCS 1717, pp.292-302, Springer-Verlag, 1999.

[CJ01] C. Clavier and M. Joye, "Universal Exponentiation Algorithm – A First Step towards Provable SPA-Resistance –", *CHES2001*, LNCS 2162, pp.300-308, Springer-Verlag, 2001.

[FGKS02] W. Fischer, C. Giraud, E. Knudsen, and J. Seifert, "Parallel Scalar Multiplication on General Elliptic Curves over $\mathbb{F}_p$ hedged against Non-Differential Side-Channel Attacks", Cryptology ePrint Archive, 2002/007, 2002. Available from `http://eprint.iacr.org/2002/007/`

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

[IT02] T. Izu and T. Takagi, "A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks", *PKC2002*, LNCS 2274, pp.280-296, Springer-Verlag, 2002.

[JQ01] M. Joye and J. Quisqiater, "Hessian Elliptic Curves and Side-Channel Attacks", *CHES2001*, LNCS 2162, pp.412-420, Springer-Verlag, 2001.

[KCJLMWY01] S. Kim, J. Cheon, M. Joye, S. Lim, M. Mambo, D. Won, and Y. Zheng, "Strong Adaptive Chosen-Ciphertext Attacks with Memory Dump (or: The Importance of the Order of Decryption and Validation)", Cryptography and Coding, 8th IMA International Conference, LNCS 2260, pp.114-127, Springer-Verlag, 2001.

[Koc96] C. Kocher, "Timing attacks on Implementations of Diffie-Hellman, RSA, DSS, and other Systems", *Crypto'96*, LNCS 1109, pp.104-113, Springer-Verlag, 1996.

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

[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.

[LS01] P. Liardet and N. Smart, "Preventing SPA/DPA in ECC system using the Jacobi Form", *CHES2001*, LNCS 2162, pp.401-411, Springer-Verlag, 2001.

[Man01] J. Manger, "A Chosen Ciphertext Attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as Standardized in PKCS #1 v2.0", *CRYPTO 2001*, LNCS 2139, pp.230-238, Springer-Verlag, 2001.

[Mon87] P. Montgomery, "Speeding the Pollard and Elliptic Curve Methods for Factorizations", *Math. of Comp*, vol.48, pp.243-264, 1987.

[NS01] P. Nguyen and I. Shparlinski, "The Insecurity of the Digital Signature Algorithm with Partially Known Nonces", to appear in the Journal of Cryptology.

[Nov02] R. Novak, "SPA-based Adaptive Chosen-Ciphertext Attack on RSA Implementation", *PKC2002*, LNCS 2274, pp.252-262, Springer-Verlag, 2002.

[OK-ECDH] Key Agreement Scheme OK-ECDH, Hitachi, 2001. Available from `http://www.sdl.hitachi.co.jp/crypto/ok-ecdh/index.html`

[OK-ECDSA] Digital Signature Scheme OK-ECDSA, Hitachi, 2001. Available from `http://www.sdl.hitachi.co.jp/crypto/ok-ecdsa/index.html`

[OKS00] K. Okeya, H. Kurumatani, and K. Sakurai, "Elliptic Curves with the Montgomery Form and their cryptographic Applications", *PKC2000*, LNCS 1751, pp.446-465, Springer-Verlag, 2000.

[Sch00] W. Schindler, "A Timing Attack against RSA with the Chinese Remainder Theorem", *CHES 2000*, LNCS 1965, pp.109-124, 2000.

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

A Numerical Examples

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

	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$	$d=9$
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	-	-	-	-	-

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

A.1 Self-collision Polynomial $f_d(x)$

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

$$
\begin{aligned}
f_2(x) &= 3x^2 + a \\
f_3(x) &= 7x^6 + 11ax^4 - 4bx^3 + 13a^2x^2 + 20abx + a^3 + 16b^2 \\
f_4(x) &= 13x^{12} + 70ax^{10} + 52bx^9 + 231a^2x^8 + 912abx^7 + (100a^3 + 1536b^2)x^6 \\
&\quad +408a^2bx^5 + (43a^4 + 1776ab^2)x^4 + (-176a^3b + 1024b^3)x^3 \\
&\quad +(54a^5 + 96a^2b^2)x^2 + (84a^4b + 448ab^3)x + a^6 + 48a^3b^2 + 256b^4 \\
f_5(x) &= 21x^{20} + 298ax^{18} + 828bx^{17} + 1917a^2x^{16} + 16224abx^{15} \\
&\quad +(-360a^3 + 43920b^2)x^{14} + 3024a^2bx^{13} + (938a^4 + 88368ab^2)x^{12} \\
&\quad +(-31200a^3b + 42432b^3)x^{11} + (11484a^5 + 42768a^2b^2)x^{10} \\
&\quad +(-600a^4b + 113600ab^3)x^9 + (13794a^6 + 26928a^3b^2 + 101376b^4)x^8 \\
&\quad +(45216a^5b + 127872a^2b^3)x^7 + (4312a^7 + 104496a^4b^2 + 252672ab^4)x^6 \\
&\quad +(16464a^6b + 169344a^3b^3 + 129024b^5)x^5 + (225a^8 + 38160a^5b^2 + 276480a^2b^4)x^4 \\
&\quad +(-1056a^7b + 28352a^4b^3 + 254976ab^5)x^3 \\
&\quad +(138a^9 - 720a^6b^2 + 768a^3b^4 + 86016b^6)x^2 + (252a^8b + 1728a^5b^3)x \\
&\quad +a^{10} + 144a^7b^2 + 1536a^4b^4 + 4096ab^6
\end{aligned}
$$

A.2 Self-collision Points

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

The following data are self-collision points on the standardized curve `secp128r1`. All data

are described in decimal.

$$\begin{aligned} p &= 340282366762482138434845932244680310783 \\ a &= 340282366762482138434845932244680310780 \\ b &= 308990863222245658030922601041482374867 \end{aligned}$$

- 3rd self-collision (2 points)

  (216271600127653182337595708199431611080, 194803870373360954129586700702766210 99)
  (216271600127653182337595708199431611080, 320801979725146043021887262174403689684)

- 4-th self-collision (4 points)

  (335239303131188189592410503550901274814, 163560355908249703608369064173582669 10)
  (335239303131188189592410503550901274814, 323926331171657168074009025827322043873)
  (540270477431855030313790089862571485 98, 297648799076421795422690158452388458743)
  (540270477431855030313790089862571485 98, 426335676860603430121557737922918520 40)

- 7-th self-collision (2 points)

  (827549507532447413390802517516141905 04, 476681622210732318747264256683414581 75)
  (827549507532447413390802517516141905 04, 292614204541408906560119506576338852608)

- 8-th self-collision (4 points)

  (199339362410482128911221164674409814519, 253221411113625015954792980829355172921)
  (199339362410482128911221164674409814519, 870609556488571224800529514153251378 62)
  (205366795446775620759856339377774510936, 206332927623175796680656223786069620681)
  (205366795446775620759856339377774510936, 133949439139306341754189708458610690102)