Side Channel Attacks and Countermeasures on Pairing Based Cryptosystems over Binary Fields

Tae Hyun Kim[1], Tsuyoshi Takagi[2],
Dong-Guk Han[3], Ho Won Kim[3], and Jongin Lim[1]

[1] Center for Information and Security Technologies(CIST),
Korea University, Seoul, Korea
{thkim, jilim}@cist.korea.ac.kr
[2] FUTURE UNIVERSITY-HAKODATE, Japan
takagi@fun.ac.jp
[3] Electronics and Telecommunications Research Institute(ETRI), Korea
{christa, khw}@etri.re.kr

Abstract. Pairings on elliptic curves have been used as cryptographic primitives for the development of new applications such as identity based schemes. For the practical applications, it is crucial to provide efficient and secure implementations of the pairings. There have been several works on efficient implementations of the pairings. However, the research for secure implementations of the pairings has not been thoroughly investigated. In this paper, we investigate vulnerability of the pairing used in some pairing based protocols against side channel attacks. We propose an efficient algorithm secure against such side channel attacks of the eta pairing using randomized projective coordinate systems for the pairing computation.

Keywords: Pairing based cryptosystems, Side channel attacks, Differential Power Analysis, Randomized projective coordinate systems, Eta pairing.

1 Introduction

Since pairings have new and useful cryptographic properties such as bilinearity and non-degeneracy the interest and active research of them in cryptography is growing. Recently many cryptographic schemes based on the Tate pairing and the Weil pairing have been proposed. For example, identity based encryption schemes [6,28], identity based signature schemes [17,8,26], short signature [7], and identity based authenticated key agreement [31].

To accelerate practical applications of pairing based schemes a lot of work has focused on the development of efficient and easy computations of pairings on elliptic curves. Barreto et al. [2] and Galbraith et al. [13] provided the fast computation of the Tate pairing on supersingular elliptic curves over finite fields of characteristic two and three. Duursma and Lee [11] gave a closed formula in the case of characteristic three, and Kwon [21] extended it to supersingular curves over characteristic two. Barreto et al. [1] proposed a general technique

D. Pointcheval, Y. Mu, and K. Chen (Eds.): CANS 2006, LNCS 4301, pp. 168–181, 2006.

for the efficient computation of pairings on supersingular abelian varieties called *the eta pairing*.

Recently such methods of pairings have been implemented in software and hardware to accelerate constrained devices such as smartcards [5,14,30,4,27]. In the implementation of cryptosystems or protocols on such devices, we should consider not only efficiency but also security. If we don't carefully implement cryptosystems on constrained devices then they can be insecure against side channel attacks (SCAs). Thus it is important to consider the secure implementation of pairing based cryptosystems secure against SCAs. We can divide pairing based schemes into two types by whether or not an input of pairing is secret [10]. For example, identity based signature schemes such as short signature scheme by Boneh et al. require the secret information as an input (i.e., the secret scalar) of the elliptic curve scalar multiplication. Side channel attacks and countermeasures on scalar multiplications have well been studied. However, identity based encryption schemes such as Boneh-Franklin encryption scheme [6] use the secret information as an input of the pairing. In this case, there are only few works of SCAs on the pairings [24,29,33]. In [24], Page and Vercauteren showed side channel attacks against the Duursma-Lee algorithm. In [29], Scott suggested countermeasures to provide resistance to more sophisticated simple power analysis (SPA) and differential power analysis (DPA) attacks. Very recently, Whelan and Scott investigated practical pairing algorithms using correlation power analysis (CPA) [33]. However the form of some multiplication used in the eta pairing on the supersingular curves in characteristic two is different to the case of characteristic three. In this paper, we concretely examine the security of the eta pairing on the supersingular curve over $\mathbb{F}_{2^m}$ against timing attack (TA) or SPA attack and DPA attack.

In general, to speed up elliptic curve point addition and doubling, the projective coordinate systems are used instead of the affine coordinate system because the affine coordinate system requires a modular inversion operation, computationally expensive. In [19], Izu and Takagi showed that the Tate pairing on general elliptic curves over prime fields $\mathbb{F}_{p^m}$ is efficiently computed using the projective coordinate systems. Hess et al. [18] extended the eta pairing over supersingular curves to general curves over prime fields $\mathbb{F}_{p^m}$, and then examined efficiency in the projective coordinate systems. However, for providing protection of SCAs, Coron [9] used the randomized projective coordinate. In this paper, to resist SCAs, we propose an explicit algorithm using randomness of the projective coordinate systems of the eta pairing for a curve over characteristic two.

This paper is organized as follows: In the next section we review several methods for the efficient computation of the Tate pairing. Section 3 describes side channel attacks on the eta pairing over supersingular curves in characteristic two. Section 4 presents a countermeasure to prevent the attack described in Section 3. Section 5 compares the proposed countermeasure with the previous methods. Finally we conclude in Section 6.

2 The Tate Pairing

Let E be an elliptic curve over a finite field $\mathbb{F}_q$. Let l be a positive integer coprime to q, which divides $\#E(\mathbb{F}_q)$, i.e., $l|\#E(\mathbb{F}_q)$. Let k be the smallest positive integer such that the l-th root of unity exists in $\mathbb{F}_{q^k}^*$, i.e., $l|(q^k-1)$. We call such k the embedding degree or security multiplier. The Tate pairing of P and Q on $E(\mathbb{F}_q)$ of order l is defined as follows:

$$e_l : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_{q^k})[l] \to \mu_l \quad \text{with} \quad e_l(P,Q) = f_{l,P}(\mathcal{D}_Q)^{(q^k-1)/l}, \tag{1}$$

where $f_{l,P}$ is a rational function such that $(f_{l,P}) = l(P) - ([l]P) - (l-1)(\mathcal{O})$ and $\mathcal{D}_Q$ is a zero divisor equivalent to $(Q) - (\mathcal{O})$ such that $\mathcal{D}_Q$ and $(f_{l,P})$ have disjoint supports. Also μ_l is the subgroup of the l-th root of unity in $\mathbb{F}_{q^k}^*$. The first efficient method for computing such a rational function is proposed by Miller [22]. This algorithm is based on the binary method for elliptic curve scalar multiplication combined with an evaluation of the tangent lines used in the elliptic curve addition process. In the original Miller algorithm, a denominator in the step of an evaluation of the tangent lines should be manipulated. Barreto et al. [2] showed the way able to speed up by eliminating the denominator, namely, for supersingular elliptic curves they used a specific endormorphism ψ called a distortion map [32]. Thus the Tate pairing is modified by

$$e_l(P,Q) = f_{l,P}(\psi(Q))^{(q^k-1)/l}. \tag{2}$$

To improve the computation speed of the above pairing on curves we can use $N = hl$ to be a multiple of the order of elliptic curve for some integer h instead of l [13]. Since $(f_{N,P}) = h(f_{l,P}) = (f_{l,P}^h)$ the Tate pairing can be computed by $f_{N,P}(\psi(Q))^{(q^k-1)/N}$, where $N = hl$ and $f_{N,P}$ is a rational function such that $(f_{N,P}) = N(P) - ([N]P) - (N-1)(\mathcal{O})$. Using this property, Duursma and Lee [11] and Kwon [21] replaced l by $N = hl$ which has low Hamming weight in the case of characteristic three and two, respectively. In characteristic two, the order and the embedding degree of supersingular curves $E : y^2 + y = x^3 + x + b$, where $b \in \mathbb{F}_2$ are $2^m \pm 2^{(m+1)/2} + 1$ and 4, respectively. Thus, we can use $N = 2^{2m}+1 = (2^m+2^{(m+1)/2}+1)(2^m-2^{(m+1)/2}+1)$ of Hamming weight 2 in the binary representation and also the final exponentiation by $(2^{4m}-1)/(2^{2m}+1) = 2^{2m}-1$ is very simple, which is computed by applying one Frobenius map and one division [11,21].

The fastest method for computing the Tate pairing is the eta pairing [1], which includes the algorithms by Duursma and Lee [11] and Kwon [21] as special cases. We now present an outline of the eta pairing algorithm. The elliptic curve of our interest is the supersingular curve $E : y^2 + y = x^3 + x + b$ over $\mathbb{F}_{2^m}$ where $m \equiv 3 \bmod 8$ and $b \in \mathbb{F}_2$. The extension field $\mathbb{F}_{2^{4m}}$ is represented by the basis $\{1, s, t, st\}$ such that $s^2 + s + 1 = 0$ and $t^2 + t + s = 0$. The distortion map is $\psi(x, y) = (x + s^2, y + sx + t)$. For some integer T the eta pairing η_T is defined to be $\eta_T(P,Q) = f_{T,P}(\psi(Q))$. Then there is the following relation between the eta pairing and the Tate pairing.

$$\big(\eta_T(P,Q)^M\big)^{aT^{a-1}} = \big(e_N(P,Q)\big)^L. \tag{3}$$

where $T^a + 1 = LN$ for some $a \in \mathbb{N}$ and $L \in \mathbb{Z}$, $T = q + cN$ for some $c \in \mathbb{Z}$, and $M = (q^k - 1)/N$. To reduce of the loop in characteristic two we can choose $T = \mp 2^{(m+1)/2} + 1$, $a = 2$, $c = -1$, and $L = 2$. In this case, we should first compute the rational function corresponding to addition of $2^{(m+1)/2}P$ and $\pm P$. Since $2^{(m+1)/2}P$ is efficiently computed by Frobenius map we can easily deal with it. We have

$$\left(\eta_T(P,Q)^M\right)^{2T} = e_N(P,Q)^2 \;\Rightarrow\; \left(\eta_T(P,Q)^M\right)^T = e_N(P,Q). \tag{4}$$

However, to obtain the same result as the Tate pairing, it must be further exponentiated to the power of T. The concrete algorithm of the eta pairing on supersingular curves in characteristic two with $m \equiv 3 \bmod 8$ is shown in Algorithm 1.

Algorithm 1. $\eta_T(P,Q)$ on the curve E: $y^2 + y = x^3 + x + b$ over $\mathbb{F}_{2^m}$, where $b \in \mathbb{F}_2$ and $m \equiv 3 \bmod 8$ case [1]

Input: $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$.
Output: $\eta_T(P,Q)$.
1: $u \leftarrow x_P + 1$
2: $f \leftarrow u \cdot (u + x_Q) + y_P + y_Q + b + 1 + (u + x_Q)s + t$
3: **for** $i = 0$ to $(m+1)/2$ **do**
4: $\quad u \leftarrow x_P$, $x_P \leftarrow \sqrt{x_P}$, $y_P \leftarrow \sqrt{y_P}$
5: $\quad g \leftarrow u \cdot (x_P + x_Q) + y_P + y_Q + x_P + (u + x_Q)s + t$
6: $\quad f \leftarrow f \cdot g$
7: $\quad x_Q \leftarrow x_Q^2$, $y_Q \leftarrow y_Q^2$
8: **end for**
9: **return** $f^{(2^{2m}-1)(2^{2m}-2^{(m+1)/2}+1)}$

Note that we mainly deal with the eta pairing in characteristic two because the algorithm is simpler in characteristic three than characteristic two in the sense of side channel attacks.

3 Side Channel Attacks

Side channel attacks (SCAs) have been recognized as serious menaces to constrained devices such as smartcards. By monitoring computation timing, power consumption, or electromagnetic radiation, etc. during cryptographic operations, it is possible to recover the secret information related to the keys inside the device [20,9]. Timing attack (TA) analyzes the time taken to execute cryptographic algorithms. Simple Power Analysis (SPA) attack directly interprets power consumption measurements collected during cryptographic operations. Differential Power Analysis (DPA) attack analyzes correlation between power consumptions and specific key-dependent intermediate values which appear during computation with the secret by using statistical tools and error correction techniques.

In this section we investigate the eta pairing used in identity based encryption schemes such as Boneh-Franklin encryption scheme [6] and Sakai-Kasahara encryption scheme [28] in the context of side channel attacks such as TA, SPA, and DPA.

3.1 Weak Point in Pairing Computation

In the decryption step of identity based encryption schemes, the critical calculation is $e(S_{\mathsf{ID}}, C)$ where S_{ID} is the fixed secret key and C is a part of a ciphertext. In this case, side channel attacks may try to extract the secret key from the pairing computation by repeatedly manipulating C. Recently, Page and Vercauteren [24] presented an SPA attack and a DPA attack on a field multiplication step of the pairing computation with the secret value. They showed that there are such field multiplications in the Duursma-Lee algorithm [24] and the BLKS algorithm [2] of characteristic three, i.e., $y \cdot r$ where y is an unknown and fixed value related with the y-coordinate of the secret point S_{ID} and r is a known and variable value related with the ciphertext C. Since the field multiplication is analogous to exponentiation on a multiplicative group or scalar multiplication on an additive group we can easily apply DPA attacks such as [9].

However, the eta pairing in characteristic two includes the multiplication of the different form $a \cdot (b+r)$ compared with the case of characteristic three, where both a and b are unknown. In this case, since r chosen by an attacker is added by the unknown value b we may not simulate or guess an intermediate value related with secret value. Thus it seems secure against DPA attacks. However, in the next section, we will show that the addition and the multiplication of $a \cdot (b + r)$ can be insecure against TA or SPA attack and DPA attack.

Assumption. From this section assume that the first input $P = (x_P, y_P)$ of the pairing in Algorithm 1 is secret and the second input $Q = (x_Q, y_Q)$ is public. Note however that the description of the attack is similar even if P is public and Q is secret. For the computation of $a \cdot (b + r)$, we also assume that the addition $(b + r)$ is first computed, and the multiplication $a \cdot (b + r)$ is computed.

3.2 Finite Field Arithmetic

Let $f(x)$ be an irreducible polynomial of degree m over $\mathbb{F}_2$. Assume an element a of $\mathbb{F}_{2^m} \simeq \mathbb{F}_2[x]/(f(x))$ is represented by the polynomial basis. Let the bit string $(a_{m-1}a_{m-2} \cdots a_1 a_0)$ where $a_i \in \mathbb{F}_2$ denote an element a of $\mathbb{F}_{2^m}$. Addition and multiplication of $a = (a_{m-1} \cdots a_1 a_0)$ and $b = (b_{m-1} \cdots b_1 b_0)$ in $\mathbb{F}_{2^m}$ are performed as follows:

Addition: $a + b = (c_{m-1} \cdots c_1 c_0)$, where $c_i = (a_i + b_i) \bmod 2$.

Multiplication: $c = a \cdot b = (c_{m-1} \cdots c_1 c_0)$, where c is computed as a multiplication of polynomials $a(x)$ and $b(x)$ of $\mathbb{F}_2[x]$ followed by a reduction by $f(x)$. That is, $c = a(x) \cdot b(x) \bmod f(x)$.

The addition of two elements a and b in $\mathbb{F}_{2^m}$ is easily performed by a bitwise XOR operation. A usual way of multiplying two elements a and b of $\mathbb{F}_{2^m}$ is done by scanning the multiplier b one bit at a time. This method is known as the shift-and-add method based on the following observation

$$a \cdot b = a_{m-1}x^{m-1}b + a_{m-2}x^{m-2}b + \cdots + a_1 xb + a_0 b.$$

For efficiency reason the irreducible polynomial is selected as a trinomial or a pentanomial. Therefore, we assume that a multiplication of polynomials is first performed, and then the result is reduced by an irreducible polynomial. Note however that the attack is not limited to the above multiplication and the separative computation of multiplication and reduction. we can extend our attack to other multiplication methods and the simultaneous computation of multiplication and reduction. The concrete algorithm of the shift-and-add method is given in Algorithm 2.

Algorithm 2. Shift-and-add(Right-to-left) method for polynomial multiplication

Input: $a(x) = (a_{m-1} \cdots a_0)_2$ and $b(x) = (b_{m-1} \cdots b_0)$.
Output: $c(x) = a(x) \cdot b(x)$.
1: $C \leftarrow 0$ and $B \leftarrow b$
2: **for** $i = 0$ to $m - 1$ **do**
3: **if** $a_i = 1$ **then**
4: $C \leftarrow C + B$
5: **end if**
6: $B \leftarrow B \cdot x$
7: **end for**
8: **return** C

3.3 SPA or Timing Attack on the Eta Pairing in Characteristic Two

We consider the multiplication $a \cdot (b + r)$. If a is multiplier then the addition is performed depending on whether $a_i = 1$ or not. The structure of the shift-and-add method for the multiplication of $a \cdot (b + r)$ is shown in Figure 1.

Page and Vercauteren [24] firstly presented an SPA attack against $a \cdot r$ in F_{3^m}, where a is secret and r is public. It means that this conditional branch is vulnerable to TA or SPA attack [20,9]. Thus, we can also recover u of $u \cdot (x_P + x_Q)$, which is a part of step 5 of Algorithm 1. In this case, u is the x-coordinate of the secret point P. Thus we can obtain the y-coordinate from the x-coordinate.

However, if $b + r$ is multiplier then the conditional branch occurs depending on $b_i + r_i$. So, if we can detect an appearance of the conditional branch by TA or SPA attack then $b_i \neq r_i$. Otherwise, $b_i = r_i$. Thus we can recover x_P by controlling x_Q in $u \cdot (x_P + x_Q)$. Since x_P is the square root of the x-coordinate of the secret point P we can obtain the x-coordinate by squaring x_P, and then the y-coordinate from the x-coordinate of the secret point P.

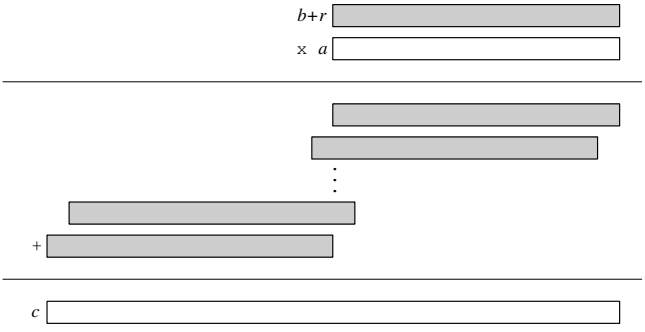

Fig. 1. The shift-and-add multiplication of $a \cdot (b + r)$ in $\mathbb{F}_{2^m}$

3.4 DPA on the Eta Pairing in Characteristic Two

In this section, we investigate DPA attacks against the addition $b + r$ and the multiplication $a \cdot (b + r)$ used in the eta pairing on curves in characteristic two, where a and b are secret and r is public.

To ease the explanation we consider the simplified Hamming weight model for power leakage [23]. In this model, power consumption depends on the Hamming weight of the data being processed. Thus we can express the power consumption $\mathcal{C}$ as follows:

$$\mathcal{C} = \varepsilon \cdot H + n,$$

where H, ε and n represent the Hamming weight of the intermediate data, the incremental amount of power for each extra '1' in the Hamming weight, and the noise, respectively. Note that assume the average of noise n is zero.

Attack on the Addition. The addition of $b + r$ is of the form

$$(b_{m-1} \oplus r_{m-1})x^{m-1} + (b_{m-2} \oplus r_{m-2})x^{m-2} + \cdots + (b_1 \oplus r_1)x + (b_0 \oplus r_0).$$

Let $\mathcal{C}$ be the power consumption associated with the addition operation $b+r$. To recover the i-th bit of b, we guess that $b_i = 1$ and divide power consumptions into two sets by r_i.

$$S_k = \{\mathcal{C} | r_i = k\} \text{ with } k \in \{0, 1\} \tag{5}$$

The averages of S_0 and S_1 are respectively $\varepsilon(M + 1)/2$ and $\varepsilon(M - 1)/2$, where M is the size of the resister. Thus the differential power consumption is

$$\Delta = \langle S_0 - S_1 \rangle. \tag{6}$$

If $\Delta \neq 0$ then $b_i = 1$, otherwise $b_i = 0$.

In conclusion, since the addition operation $(x_P + x_Q)$ of step 5 of Algorithm 1 is vulnerable to the above analysis we can recover x_P. From this value we can obtain the x-coordinate and y-coordinate of the secret point P.

Attack on the Multiplication. Since DPA attacks use correlation between power consumptions and specific key-dependent intermediate values which appear during computation related with the secret it is important to examine the intermediate values for the computation of $a \cdot (b + r)$ by Algorithm 2. We will treat both cases: the first case is that a is multiplier and $b + r$ is multiplicand and the second case is that $b + r$ is multiplier and a is multiplicand.

Theorem 1. *In Algorithm 2, if a is multiplier and $b + r$ is multiplicand then the algorithm is vulnerable to the DPA attack.*

Proof. In this case, the multiplication is performed as follows;

$$a_{m-1}x^{m-1}(b+r) + a_{m-2}x^{m-2}(b+r) + \cdots + a_1x(b+r) + a_0(b+r). \tag{7}$$

To describe how to recover a assume the lowest bits $a_{i-1}, \cdots, a_0$ of the secret multiplier a are already recovered. We describe how to find the next bit a_i.

In Algorithm 2, the intermediate value obtained at end of i-th step of the loop is

$$a_ix^i(b+r) + \cdots + a_1x(b+r) + a_0(b+r). \tag{8}$$

The i-th bit of (8) is

$$\begin{aligned} & a_i(b_0 + r_0) + a_{i-1}(b_1 + r_1) + \cdots + a_1(b_{i-1} + r_{i-1}) + a_0(b_i + r_i) \\ = \ & (a_ib_0 + a_{i-1}b_1 + \cdots + a_0b_i) + (a_ir_0 + a_{i-1}r_1 + \cdots + a_0r_i). \end{aligned} \tag{9}$$

In this case, we can not simulate this intermediate value because we don't know the value of $a_ib_0 + a_{i-1}b_1 + \cdots + a_0b_i$. However, we can accomplish a DPA attack by only controlling the input value r and not the intermediate value.

Let $\mathcal{C}$ be the power consumption associated with computation of $a \cdot (b + r)$. From the formula of (9) we guess the $a_i = 1$ and divide power consumptions into two sets by $a_ir_0 + a_{i-1}r_1 + \cdots + a_0r_i$, which is derived from the already known $a_{i-1}, \cdots, a_0$ and the random value chosen by ourself.

$$S_k = \{\mathcal{C} | r_0 + a_{i-1}r_1 + \cdots + a_0r_i = k\} \text{ with } k \in \{0, 1\} \tag{10}$$

If $a_ib_0 + a_{i-1}b_1 + \cdots + a_0b_i = 1$ in (9) then the average power consumptions of S_0 and S_1 are respectively $\varepsilon(M+1)/2$ and $\varepsilon(M-1)/2$, where M is the size of the resister. So, we have $\langle S_1 - S_0 \rangle = -\varepsilon$. In the case of $a_ib_0 + a_{i-1}b_1 + \cdots + a_0b_i = 0$, the averages of S_0 and S_1 are respectively $\varepsilon(M-1)/2$ and $\varepsilon(M+1)/2$. We have $\langle S_1 - S_0 \rangle = \varepsilon$. The difference between two cases is only whether the differential average power consumptions is positive or negative. Thus we will compute the differential power consumption by using absolute value

$$\Delta = \langle |S_1 - S_0| \rangle. \tag{11}$$

If $\Delta \neq 0$, namely, we can detect an appreciable peak then the guess is right (i.e., $a_i = 1$), otherwise the guess is wrong (i.e., $a_i = 0$). The remaining bits $a_{m-1}, \cdots, a_{i+1}$ are recursively recovered by the same way. Thus we can recover the multiplier of Algorithm 2. □

Theorem 2. *In Algorithm 2, if $b + r$ is multiplier and a is multiplicand then the algorithm is vulnerable to the DPA attack.*

Proof. The multiplication of $(b + r) \cdot a$ is performed as follows;

$$(b_{m-1} + r_{m-1})x^{m-1}a + (b_{m-2} + r_{m-2})x^{m-2}a + \cdots + (b_1 + r_1)xa + (b_0 + r_0)a.$$

In this case, we also describe how to recover a. Assume the lowest bits $a_{i-1}, \cdots, a_0$ of a are already recovered. The aim is to find the next bit a_i.

In Algorithm 2, the intermediate value obtained at end of i-th step of the loop is

$$(b_i + r_i)x^i a + \cdots + (b_1 + r_1)xa + (b_0 + r_0)a. \tag{12}$$

The i-th bit of (12) is

$$\begin{aligned} & (b_i + r_i)a_0 + (b_{i-1} + r_{i-1})a_1 + \cdots + (b_1 + r_1)a_{i-1} + (b_0 + r_0)a_i \\ = \; & (b_i a_0 + b_{i-1}a_1 + \cdots + b_0 a_i) + (r_i a_0 + r_{i-1}a_1 + \cdots + r_0 a_i). \end{aligned} \tag{13}$$

The above equation is equal to (9). Thus, we can recover a_i by the proof of Theorem 1. □

In conclusion, since the multiplication of $u(x_P + x_Q)$ of step 5 of Algorithm 1 is vulnerable to the proposed attack, we can recover u, the x-coordinate of the secret point P. Finally, we can obtain the y-coordinate from the x-coordinate.

Remark 1. Since the eta pairing over characteristic three also includes the addition $a + r$ and the multiplication $a \cdot r$, where a is secret and r is public (See [1] for detail.) the addition and the multiplication are vulnerable to the above described attack. However, note that Page and Vercauteren [24] also presented DPA attack on the multiplication of $a \cdot r$. Thus the eta pairing over characteristic three is also insecure against side channel attacks.

4 Proposed Countermeasure

The attack described in Section 3 is possible since we can choose and control an input value. To incapacitate such a behavior Coron [9] proposed three methods for securely computing scalar multiplication dP in elliptic curve cryptosystems (ECCs), where d is the secret key and P is public. Let E be an elliptic curve over finite fields $\mathbb{F}_q$. Let $\#E$ be the number of points of the curve. The countermeasures for computing $Q = dP$ are follows:

1. Randomization of the private value, i.e., $dP = (d + r \cdot \#E)P$ for a random number $r \in \mathbb{F}_q$.
2. Blinding the public value, i.e., $dP = d(P + R) - dR$ for a random point R on $E(\mathbb{F}_q)$.
3. Randomized projective coordinate, i.e., $(X; Y; Z) = (\lambda X; \lambda Y; \lambda Z)$ for a random value $\lambda \neq 0$ in $\mathbb{F}_q$.

In the context of the above techniques, Page and Vercauteren [24] and Scott [29] proposed some methods for the pairings. First, they used bilinearity to randomize the private data, i.e., $e(P,Q) = e(sP, tP)^{1/st}$ where s and t are random variables. Furthermore, the exponentiation to the power $1/st$ can be removed by selecting s and t satisfying $s \cdot t = 1 \bmod l$, where l is the order of the underlying elliptic curve for the pairing [24]. Second, they presented the method for blinding the input point by the relation $e(P,Q) = e(P, Q+R) \cdot e(P,R)^{-1}$ [24].

4.1 Projective Coordinate Randomization

In this paper, we propose an explicit algorithm for the eta pairing using the projective coordinate in order to resist DPA attack, which is the fastest method among existing countermeasures, and then we estimate computational efficiency between the proposed method and previous countermeasures. We now describe how to make an algorithm using the randomized projective coordinate technique of the eta pairing in the case of characteristic two. The projective coordinates (X, Y, Z) of a point $P = (x, y)$ are given by

$$x = X/Z,\ y = Y/Z$$

We can randomize the input points by randomly selecting Z-coordinate value before the computation of the pairing $e(P,Q)$. In Algorithm 1, we only randomize Q for efficiency reason. Thus the step 5 of Algorithm 1 for the eta pairing is changed into

$$\begin{aligned} & x_P(\sqrt{x_P} + X_Q/Z_Q) + \sqrt{y_P} + \sqrt{x_P} + Y_Q/Z_Q + (x_P + X_Q/Z_Q)s + t \\ = \ & \frac{1}{Z_Q}\Big(x_P(Z_Q\sqrt{x_P} + X_Q) + Z_Q(\sqrt{y_P} + \sqrt{x_P}) + Y_Q + (Z_Q x_P + X_Q)s + Z_Q t\Big). \end{aligned}$$

Since $1/Z_Q$ is in $\mathbb{F}_q$ it becomes one after the final exponentiation. Thus elimination of $1/Z_Q$ does not effect the result. The concrete algorithm is shown in Algorithm 3.

The step 6 of the proposed countermeasure is secure against the attacks described in the previous section because all operands of the addition and the multiplication operations are randomized by the projective coordinate system.

4.2 Randomizing Miller Variables

In [29], Scott introduced a method which multiplies intermediate values appearing during the loop by a random value in $\mathbb{F}_q$. To defend from the DPA attacks described the previous section, all intermediate variables (not only g) in step 2 and the step 5 of Algorithm 1 must be multiplied by a random value. Thus the step 2 and the step 5 should be respectively changed into $f \leftarrow u \cdot (r \cdot u + r \cdot x_Q) + r \cdot y_P + r \cdot (y_Q + b + 1) + (r \cdot u + r \cdot x_Q)s + t$ and $g \leftarrow u \cdot (r \cdot x_P + r \cdot x_Q) + r \cdot y_P + r \cdot y_Q + r \cdot x_P + (r \cdot u + r \cdot x_Q)s + rt$.

Algorithm 3. Randomized Projective Coordinate $\eta_T(P,Q)$ on the curve E: $Y^2Z + YZ^2 = X^3 + XZ^2 + bZ^3$ over $\mathbb{F}_{2^m}$, where $b \in \mathbb{F}_2$ and $m \equiv 3 \bmod 8$ case

Input: $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$.
Output: $\eta_T(P,Q)$.
1: $(X_Q, Y_Q, Z_Q) \leftarrow (\lambda x_Q, \lambda y_Q, \lambda)$, where λ is a random integer.
2: $u \leftarrow x_P + 1$
3: $f \leftarrow u \cdot (Z_Q \cdot u + X_Q) + Z_Q \cdot (y_P + b + 1) + Y_Q + (Z_Q \cdot u + X_Q)s + Z_Q t$
4: **for** $i = 0$ to $(m+1)/2$ **do**
5: $\quad u \leftarrow x_P$, $x_P \leftarrow \sqrt{x_P}$, $y_P \leftarrow \sqrt{y_P}$
6: $\quad g \leftarrow u \cdot (Z_Q \cdot x_P + X_Q) + Z_Q \cdot (y_P + x_P) + Y_Q + (Z_Q \cdot u + X_Q)s + Z_Q t$
7: $\quad f \leftarrow f \cdot g$
8: $\quad X_Q \leftarrow X_Q^2$, $Y_Q \leftarrow Y_Q^2$, $Z_Q \leftarrow Z_Q^2$
9: **end for**
10: **return** $f^{(2^{2m}-1)(2^{2m}-2^{(m+1)/2}+1)}$

5 Efficiency Comparison

We first estimate the computational cost of the proposed algorithm. In the original eta pairing, the initial step requires 1 multiplication in $\mathbb{F}_{2^m}$ and each loop requires 7 multiplications in $\mathbb{F}_{2^m}$, where 1 multiplication to compute g and 6 multiplications at step 6 because of the sparse form of g. However, the initial step of the proposed algorithm requires 5 multiplications in $\mathbb{F}_{2^m}$ where 2 multiplications at step 1 and 3 multiplications at step 3, and each loop requires 13 multiplications in $\mathbb{F}_{2^m}$, where 4 multiplications at step 6 and 9 multiplications at step 7. See Appendix for detail. Since addition and squaring in $\mathbb{F}_{2^m}$ are relatively inexpensive compared to multiplication and inversion we can ignore the cost of field additions and squarings [16]. Moreover, we can ignore the cost of square roots because the method described in [12] for computing square roots in $\mathbb{F}_{2^m}$ is as fast as squaring. Thus the additional cost for the eta pairing including the initial step is $3(m+1)+4$ multiplications in $\mathbb{F}_{2^m}$.

We now compare computational efficiency of the techniques described in Section 4. First, the method using bilinearity additionally requires 2 scalar multiplications. In supersingular curves, doubling a point is free and adding two distinct points requires two multiplications and one inversion [22]. In general, the ratio of inversion to multiplication is approximately 10 to 1 [16]. If we use the binary method for scalar multiplication then the additional cost of the eta pairing using this method is 12 multiplications in $\mathbb{F}_{2^m}$. Second, the method of $e(P,Q) = e(P,Q+R)e(P,R)^{-1}$ additionally requires 1 pairing computation, 1 extension field multiplication, and 1 extension field inversion. The computational cost of the eta pairing, i.e., Algorithm 1, is approximately $7(m+1)/2$ multiplications in $\mathbb{F}_{2^m}$ plus the final exponentiation required 3 applications of 2^m-Frobenius map, 4 multiplications in the extension field $\mathbb{F}_{2^{4m}}$, and 1 inversion in $\mathbb{F}_{2^{4m}}$ [1]. In the approach randomizing intermediate variables by Scott [29], the additional cost for the eta pairing is $4(m+1)+4$ multiplications in $\mathbb{F}_{2^m}$

Table 1. Additional Cost for DPA Resistance over the Eta pairing (not over the Tate pairing) on Supersingular Curves on $\mathbb{F}_{2^m}$

Countermeasure	Additional Cost
Page-Vercauteren (randomized private value) [24]	$12mM$
Page-Vercauteren (blinding public value) [24]	$3.5(m+1)M + \alpha$
Scott (randomizing intermediate value) [29]	$4(m+1)M + 4M$
The Proposed Method (Algorithm 3)	$\mathbf{3(m+1)M + 4M}$

as Section 4.2. We give a comparison table of the number of operations among existing techniques in Table 1, where M and α denote the computation time of a multiplication in $\mathbb{F}_{2^m}$ and the final exponentiation plus 1 extension field multiplication in $\mathbb{F}_{2^{4m}}$ and 1 extension field inversion in $\mathbb{F}_{2^{4m}}$, respectively.

6 Conclusion

In this paper, we have investigated the security of pairing based cryptosystems against side channel attacks. Since pairing has different properties from primitives of traditional cryptosystems such as RSA or ECC the application of pairings such as identity based schemes has been interesting for implementing on constrained devices such as smartcards. Although the work for efficient implementation of pairings has been concentrated the secure implementation has not been worked precisely. In this paper, we have investigated security against side channel attacks for implementations of the eta pairing on supersingular curves in characteristic two. To avoid such attacks we have proposed an explicit algorithm of the eta pairing using the projective coordinate and showed that the proposed method is the most efficient countermeasure compared with previous techniques.

Acknowledgments

Tae Hyun Kim and Jongin Lim were supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment).

References

1. P.S.L.M. Barreto, S. Galbraith, C. OhEigeartaigh and M. Scott, "Efficient Pairing Computation on Supersingular Abelian Varieties," Preprint 2005, to appear in Designs, Codes and Cryptography.
2. P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *CRYPTO 2002*, LNCS 2442, pp.354-368, 2002.
3. P.S.L.M. Barreto, B. Lynn, M. Scott, "Efficient implementation of pairing based cryptosystems," *Journal of Cryptology*, Vol.17, No.4, pp.321-334, 2004.

4. G. Bertoni, L. Breveglieri, P. Fragneto, and G. Pelosi, "Parallel Hardware Architectures for the Cryptographic Tate Pairing," Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06), pp.186-191, 2006.
5. G.M. Bertoni, L. Chen, P. Fragneto, K.A. Harrison, and G. Pelosi, "Computing tate pairing on smartcards," 2005. http://www.st.com/stonline/products/families/smartcard/ches2005_v4.pdf
6. D. Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing," *SIAM J. of Computing*, Vol.32, No.3, pp.586-615, 2003.
7. D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Asiacrypt 2001*, LNCS 2248, pp.514-532, 2002.
8. J.C. Cha and J.H. Cheon, "An Indentity-Based Signature from Gap Diffie-Hellman Groups," *PKC 2003*, LNCS 2567, pp.18-30, 2003.
9. J.S. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems," *CHES 1999*, LNCS 1717, pp.292-302, 1999.
10. R. Dutta, R. Barua, and P. Sarkar, "Pairing-Based Cryptographic Protocols : A Survey," Cryptology ePrint Archive, Report 2004/064, 2006. http://eprint.iacr.org/2004/064.
11. I. Duursma and H.-S. Lee, "Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$," *Asiacrypt 2003*, LNCS 2894, pp.111-123, 2003.
12. K. Fong, D. Hankerson, Julio López, and A. Menezes, "Field inversion and point halving revisited," Technical Report CORR 2003-18, University of Waterloo, August 2002.
13. S.D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," *ANTS V*, LNCS 2369, pp.324-337, 2002.
14. Gemplus, "ID based Cryptography and Smartcards," 2005. http://www.gemplus.com/smart/rd/publications/pdf/Joy05iden.pdf.
15. R. Granger, D. Page, and M. Stam, "Hardware and software normal basis arithmetic for pairing based cryptography in characteristic three," *IEEE Transactions on Computers*, Vol.54, No.7, pp.852-860, July 2005.
16. D. Hankerson, J.L. Hernandez, and A. Menezes, "Software Implementation of Elliptic Curve Cryptography over Binary Fields," *CHES 2000*, LNCS 1965, pp.1-24, 2000.
17. F. Hess, "Exponent group signature schemes and efficient identity based signature schems based on pairing," *SAC 2002*, LNCS 2595, pp.310-324, 2002.
18. F. Hess, N. Smart, and F. Vercauteren, "The eta pairing revisited," Cryptology ePrint Archive, Report 2006/110, 2006. http://eprint.iacr.org/2006/110.
19. T. Izu and T. Takagi, "Efficient Computations of the Tate Pairing for the Large MOV Degrees," *ICISC 2002*, LNCS 2587, pp.283-297, 2003.
20. C. Kocher, J. Jaffe, B. Jun,"Differential Power Analysis," *CRYPTO 1999*, LNCS 1666, pp.388-397, 1999.
21. S. Kwon, "Efficient Tate Pairing Computation for Elliptic Curves over Binary Fields,"*ACISP 2005*, LNCS 3574, pp.134-145, 2005.
22. A. Menezes, "Elliptic Curve Public Key Cryptosystems," Kluwer Academic Publishers, 1993.
23. T.S. Messerges, "Using Second-Order Power Analysis to Attack DPA Resistant Software," *CHES 2000*, LNCS 1965, pp.238-251, 2000.
24. D. Page and F. Vercauteren, "Fault and Side-Channel Attacks on Pairing Based Cryptography," Cryptology ePrint Archive, Report 2005/283, 2005. http://eprint.iacr.org/2005/283.

25. D. Page and F. Vercauteren, "A Fault Attack on Pairing Based Cryptography," To appear in IEEE Transactions on Computers 2006.
26. K.G. Paterson, "ID-based signature from pairings on elliptic curves," *Electronics Letters*, Vol.38, No.18, pp.1025-1026, 2002.
27. R. Ronan, C. OhEigeartaigh, C. Murphy, M. Scott, T. Kerins, and W. Marnane, "An Embedded Processor for a Pairing-Based Cryptosystem," Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06), pp.192-197, 2006.
28. R. Sakai and M. Kasahara, "ID based cryptosystems with pairing on elliptic curve," Cryptography ePrint Archive, Report 2003/054, 2003. `http://eprint.iacr.org/2003/054`.
29. M. Scott, "Computing the Tate Pairing," *CT-RSA 2005*, LNCS 3376, pp.293-304, 2005.
30. M. Scott, N. Costigan, and W. Abdulwahab, "Implemetation Cryptographic Pairings on Smartcards," Cryptography ePrint Archive, Report 2006/144, 2006. `htt://eprint.iacr.org/2006/144`.
31. N.P. Smart, "An identity based authentication key agreement protocol based on pairing," *Electronics Letters*, Vol.38, No.13, pp.630-632, 2002.
32. E. Verheul, "Evidence that XTR is more secure than supersingular elliptic curve cryptosystems," *Journal of Cryptology*, Vol.17, No.4, pp.277-296, 2004.
33. C. Whelan and M. Scott, "Side Channel Analysis of Practical Pairing Implementations: Which Path is More Secure?," Cryptography ePrint Archive, Report 2006/237, 2006. `htt://eprint.iacr.org/2006/237`.

A Multiplication in $F_{2^{4m}}$

The extension field $\mathbb{F}_{2^{4m}}$ is represented by the basis $\{1, s, t, st\}$ such that $s^2 + s + 1 = 0$ and $t^2 + t + s = 0$. Let $g = (g_0, g_1, g_2, g_3) = g_0 + g_1 s + g_2 t + g_3 st$ and $f = (f_0, f_1, f_2, f_3) = f_0 + f_1 s + f_2 t + f_3 st$. Then we have $h = (h_0, h_1, h_2, h_3) = f \cdot g$ where

$$
\begin{aligned}
h_0 &= f_0 g_0 + f_1 g_1 + f_3 g_2 + f_2 g_3 + f_3 g_3, \\
h_1 &= f_1 g_0 + f_0 g_1 + f_1 g_1 + f_2 g_2 + f_3 g_2 + f_2 g_3, \\
h_2 &= f_2 g_0 + f_3 g_1 + f_0 g_2 + f_2 g_2 + f_1 g_3 + f_3 g_3, \\
h_3 &= f_3 g_0 + f_2 g_1 + f_3 g_1 + f_1 g_2 + f_3 g_2 + f_0 g_3 + f_1 g_3 + f_2 g_3 + f_3 g_3.
\end{aligned}
$$

In Algorithm 3, since $g_3 = 0$ the above formula is simplified as follows.

$$
\begin{aligned}
h_0 &= f_0 g_0 + f_1 g_1 + f_3 g_2, \\
h_1 &= f_1 g_0 + f_0 g_1 + f_1 g_1 + f_2 g_2 + f_3 g_2, \\
h_2 &= f_2 g_0 + f_3 g_1 + f_0 g_2 + f_2 g_2, \\
h_3 &= f_3 g_0 + f_2 g_1 + f_3 g_1 + f_1 g_2 + f_3 g_2.
\end{aligned}
$$

Applying the Karatsuba multiplication method we can compute it by 9 multiplications.