# Construction of Efficient and Secure Pairing Algorithm and its Application

Dooho Choi, Dong-Guk Han, and Howon Kim

Electronics and Telecommunications Research Institute(ETRI),
Daejeon, KOREA
{dhchoi,christa,khw}@etri.re.kr

**Abstract.** The randomized projective coordinate (RPC) method applied to a pairing computation algorithm is a good solution that provides an efficient countermeasure against side channel attacks. In this study, we investigate measures for increasing the efficiency of the RPC-based countermeasures and construct a method that provides an efficient RPC-based countermeasure against side channel attacks. We then apply our method to the well-known $\eta_T$ pairing algorithm over binary fields and obtain an RPC-based countermeasure for the $\eta_T$ pairing; our method is more efficient than the RPC method applied to the original $\eta_T$ pairing algorithm.

**Keywords:** *Side Channel Attacks, Differential Power Analysis, Randomized Projective Coordinate, Tate Pairing, Eta Pairing*

## 1 Introduction

Pairings on elliptic curves are now well-known subject on the cryptographic area, because the pairings have been applied at many cryptographic schemes, for example, identity-based encryption [4, 26], identity-based signature [6, 12, 25], tripartite key agreement [14], short signature [5], identity-based authentication key agreement [28]. Incidentally, the pairings on elliptic curves were firstly introduced as cryptanalytic tools in [21, 8].

Since the main difficulty for efficiently implementation of the pairing based cryptographic schemes is the computation of pairing, there has been much development on computations of pairings. Barreto *et al.* [2] and Galbraith *et al.* [10] provided techniques for efficient computations of the pairings by removing the unnecessary computations from the original Miller's algorithm [22]. Duursma and Lee [7] found a closed formula of the Tate pairing over field with characteristic three, and Kwon [18] also gave a closed formula over field with characteristic two. To shorten the main loop of the Tate pairing computation, Barreto *et al.* [1] defined the Eta pairing on some supersingular curves, and more generally, Hess *et al.* [13] extended it to the Ate pairing on the non-supersingular elliptic curves.

Side channel attacks (SCA) commonly utilize a relation between side channel information related to a secret and internal values during cryptographic operations [16, 17]. For the SCA on the pairing computing algorithm, there has been a

little progress by the works of Page and Vercauteren [23], Whelan and Scott [29], and Kim *et al.* [15]. In [15], Kim *et al.* investigated security of the $\eta_T$ pairing over binary fields in context of side channel attacks.

A number of countermeasures have already been anticipated to protect pairing algorithms against SCA [23, 27, 29, 15]. In [23], the bilinearity of pairing is utilized to blind the secret point. Scott [27] proposed a very simple idea that is to multiply the Miller variable $m$ in BKLS algorithm [2] by a random element which will be eliminated in the final exponentiation. In [29], it is remarked that random value must not only be multiplied by the Miller variable, but must be multiplied by all intermediate values that make up the Miller variable in order for the countermeasure to be effective. In [15], Kim *et al.* directly applied randomized projective coordinate (RPC) method on the original Barreto *et al.*'s $\eta_T$ pairing algorithm [1], and they showed that their countermeasure is the most efficient among all existing countermeasures.

In this paper, for a given extension field equation we first provide a measurement to estimate the computation cost of RPC applied extension field equation, and we propose a method constructing an efficient and secure pairing algorithm from a given pairing algorithm. As it application, we present an efficient RPC based countermeasure of the $\eta_T$ pairing over binary field which reduces the additional computation cost by 17%, compared with Kim *et al.*'s countermeasure in [15].

This paper is organized as follows. In Section 2, we briefly introduce the definitions of the Tate and $\eta_T$ pairings, and Section 3 describes the SCAs and its countermeasures on the pairing algorithms. Section 4 is devoted to finding a measurement to estimate efficiencies of the RPC based countermeasures, and proposing a construction method of an efficient countermeasure against DPA attack. In Section 5, we apply our construction method to the well-known $\eta_T$ pairing algorithm over binary fields. In the last section, we conclude this paper.

## 2 The Pairings on Elliptic Curves

Let $E$ be an elliptic curve over a finite field $\mathbb{F}_q$, $l$ a positive integer coprime to $q$, which divides $\#E(\mathbb{F}_q)$, and $k$ the smallest positive integer such that $l|(q^k-1)$(it is called the *embedding degree*). The Tate pairing of order $l$ is defined as follows:

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

where $f_{l,P}$ is a rational function such that its principal divisor $(f_{l,P})$ is equivalent to $l(P)-(lP)-(l-1)(\mathcal{O})$ and $\mathcal{D}_Q$ is a zero divisor equivalent to $(Q)-(\mathcal{O})$ such that $\mathcal{D}_Q$ has disjoint support with $(f_{l,P})$, and $\mu_l$ is the group of the $l$-th roots of unity in $\mathbb{F}_{q^k}^*$. The Tate pairing can be basically computed by the following Miller's formula [22, 2]

$$f_{(a+b),P}(\mathcal{D}_Q) = f_{a,P}(\mathcal{D}_Q) \cdot f_{b,P}(\mathcal{D}_Q) \cdot \ell_{aP,bP}(\mathcal{D}_Q)/\upsilon_{(a+b)P}(\mathcal{D}_Q),$$

where $\ell_{aP,bP}$ is a line through points $aP$ and $bP$(it is a tangent line at $aP$ if $a=b$), and $\upsilon_{(a+b)P}$ is a vertical line at the point $(a+b)P$.

Barreto *et al.* [2] showed that $\tau_l(P,Q) = f_{l,P}(Q)^{(q^k-1)/l}$, since $l|\#E(\mathbb{F}_q)$ and $k$ is the embedding degree, and also they proved that for some supersingular curves with embedding degree $k = 2, 4, 6$, the vertical line evaluation part $v_{(a+b)P}(Q)$ can be omitted in the Miller's algorithm by using a distortion map $\psi$ from $E(\mathbb{F}_q)$ to $E(\mathbb{F}_{q^k})$. Their modified Tate pairing is as follows:

$$\hat{\tau}_l \colon E(\mathbb{F}_q)[l] \times E(\mathbb{F}_q)[l] \to \mu_l \text{ by } (P,Q) \mapsto \tau_l(P, \psi(Q)).$$

It can be directly computed that $f_{N,P}(\psi(Q))^{(q^k-1)/N} = f_{l,P}(\psi(Q))^{(q^k-1)/l}$, where $N = hl$ for some integer $h$, and it was firstly noted by Galbraith *et al.* [10]. In [7, 18], Duursma and Lee, and Kwon found the closed formulas of the Tate pairing for characteristic three and two, respectively. Their formulas were induced by computing $f_{N,P}(\psi(Q))^{(q^k-1)/N}$, and therefore they gave the efficient Tate pairing algorithms using these formulas. In case of characteristic two(resp. three), Kwon [18](resp. Duursma and Lee [7]) used $N = 2^{2m} + 1$(resp. $N = 3^{3m} + 1$).

To shorten the main loop of the pairing algorithm, Barreto *et al.* [1] defined the $\eta_T$ pairing for some supersingular curves as follows:

$$\eta_T(P, \psi(Q)) = f_{T,P}(\psi(Q))^W,$$

where $T = q \mod l$ and $W = (q^k-1)/N$, $N$ is an integer such that $l|N, N|q^k-1$, $T^a - 1 = LN$ for some $a$, $L$, and $l \nmid L$. The bilinearity and non-degeneracy of the $\eta_T$ pairing can be ensured by the following property(see [1, 13] for more details):

$$\tau_l(P, \psi(Q))^L = \eta_T(P, \psi(Q))^{aT^{a-1}}.$$

More generally, Hess *et al.* [13] extended it to the Ate pairing on the non-supersingular elliptic curves.

## 3 Side Channel Attacks

Side channel attacks (SCA) commonly utilize a relation between side channel information related to a secret and internal values during cryptographic operations [16, 17]. An attacker utilizes side channel information such as computation timing, power consumption, and electromagnetic radiation for confirming his/her guess at the secret. The attack aims at guessing the secret value (or some related information) stored at the target device. If an attacker is allowed to observe the side channel information only a few times and directly interprets them, it is called the *simple power analysis* (SPA). If the attacker can analyze the side channel information several times using a statistical tool, it is called the *differential power analysis* (DPA). The standard DPA utilizes the correlation function that can distinguish whether a specific bit is related to the observed calculation. Especially, if the time information taken to execute cryptographic algorithms is utilized, then it is called the *timing attack* (TA).

Although SCAs and countermeasures have been becoming increasingly well understood, the current emphasis in terms of asymmetric key schemes is mainly on RSA, ECC, and XTR [19].

Recently, newer primitives such as pairing algorithms have received some investigation. Firstly, Page and Vercauteren proposed fault and SCA against the Duursma-Lee algorithm [23]. Very recently, Whelan and Scott investigated practical pairing algorithms such as Tate, Eta, and Ate pairing using correlation power analysis (CPA) [29] and Kim *et al.* investigated security of the $\eta_T$ pairing over binary fields in context of side channel attacks [15].

A number of countermeasures have already been anticipated to protect pairing algorithms against SCA [23, 27, 29, 15]. In [23], the bilinearity of pairing is utilized to blind the secret point. A pairing is calculated as $\tau_l(P,Q) = \tau_l(aP, bQ)^{1/ab}$ where $a$ and $b$ are random values or $\tau_l(P,Q) = \tau_l(P, Q+R)/\tau_l(P,R)$ where $R$ is a random point. Note that the first results in an additional factor $a \cdot b$ in the exponent of the result, it can be eliminated by careful selection of $a$ and $b$ such that $a \cdot b \equiv 1 \pmod{l}$. Scott proposed a very simple idea that is to multiply the Miller variable $m$ in BKLS algorithm [2] by a random element which will be eliminated in the final exponentiation [27]. In [29], it is remarked that random value must not only be multiplied by the Miller variable, but must be multiplied by all intermediate values that make up the Miller variable in order for the countermeasure to be effective. Kim *et al.* introduced efficient and secure algorithms of the $\eta_T$ pairing using RPC systems for computing the pairing [15].

## 4 Construction of Efficient Countermeasure of Pairing Algorithm against DPA

### 4.1 Motivation

In [15], Kim *et al.* proposed the RPC method to protect DPA attack on $\eta_T$ pairing over the binary field, and showed their method is the fastest method among existing countermeasures by estimating computational cost of all proposed methods [23, 27]. Therefore the RPC method can be a good starting point for the construction of an efficient and secure pairing algorithm. In the RPC based countermeasure of the pairing algorithm against DPA attack, since the inputs of the pairing algorithm are two points $P = (\alpha, \beta)$ and $Q = (x, y)$ on the elliptic curve, there are the following three possibilities to randomize a given pairing algorithm:

1. The point $P$ is randomized as a projective coordinate $(\bar{\alpha}, \bar{\beta}, \bar{\gamma}) = (\bar{\gamma}\alpha, \bar{\gamma}\beta, \bar{\gamma})$, where $\bar{\gamma} \in \mathbb{F}_q^*$.
2. The point $Q$ is randomized as a projective coordinate $(\bar{x}, \bar{y}, \bar{z}) = (\bar{z}x, \bar{z}y, \bar{z})$, where $\bar{z} \in \mathbb{F}_q^*$.
3. Both two points $P$ and $Q$ are randomized simultaneously.

We do not need any consideration about the third method because it gives an inefficient algorithm. In order to make an efficient RPC based countermeasure against DPA, the following two problems we have to resolve now arise:

– Among the above two RPC methods, select one method that induces more efficient RPC based countermeasure.
– More extensively, find a modification of the original pairing algorithm to reduce the computational cost of the RPC based countermeasure.

The drastically increasing part of the computation cost is mainly caused by the modification of the equation over the extension field in the main loop of the pairing algorithm. Therefore the equation over the extension field needs to be examined in detail now. In the next section, we carefully investigate a special equation over the extension field for given four inputs in the base field.

### 4.2 RPC Method on Balanced Forms

Suppose that $f(\alpha, \beta; x, y)$ is a polynomial over a finite field $\mathbb{F}_q$ for given two variable pairs $(\alpha, \beta)$ and $(x, y)$, and $f_{(\alpha,\beta)}$ denotes the rearranged $f(\alpha, \beta; x, y)$ as an $(\alpha, \beta)$-variable polynomial. Similarly, $f_{(x,y)}$ means the rearranged $f(\alpha, \beta; x, y)$ as an $(x, y)$-variable polynomial.

**Definition 1.** *$f_{(\alpha,\beta)}$ is called a balanced form over $\mathbb{F}_q$ if it is represented as follows:*

$$f_{(\alpha,\beta)} := \sum_{i=n}^{1} (h_i(x, y)\alpha^{e_i} + g_i(x, y)\beta^{e_i}) + c_0(x, y) \tag{1}$$

*such that*

– *$e_n > e_{n-1} > \cdots > e_1 \in \mathbb{Q} \setminus \{0\}$ and*
– *for $i = 1, \ldots, n$, $h_i(x, y)$ and $g_i(x, y)$ are not zeros simultaneously.*

*Furthermore, if $f_{(\alpha,\beta)}$ and $f_{(x,y)}$ are both balanced forms then $f(\alpha, \beta; x, y)$ is said to be a balanced form.*

In Definition 1, $h_i(x, y)$ and $g_i(x, y)$ are called *coefficient polynomials*, $n$ and $e_n$ are called an *index* and *degree* of $f_{(\alpha,\beta)}$ respectively, and each $h_i(x, y)\alpha^{e_i}$ or $g_i(x, y)\beta^{e_i}$ is called a *term* of $f_{(\alpha,\beta)}$. Note that the index and degree are regarded as 0 in case of $f_{(\alpha,\beta)} = 0$ or 1. More explicitly the index of $f_{(\alpha,\beta)}$ is defined as follows:

**Definition 2.** *Suppose that $f(\alpha, \beta; x, y)$ is a balanced form and $f_{(\alpha,\beta)}$ is represented as in (1). Then*

– *The index of $f_{(\alpha,\beta)}$ is defined as $n - 1$ if the constant term $c_0(x, y)$ is zero or one, and $n$ otherwise.*
– *The index of $f_{(\alpha,\beta)}$ is defined as the index of $f'_{(\alpha,\beta)}$ if $f_{(\alpha,\beta)} = (f'_{(\alpha,\beta)})^{e'}$, for some integer $e'$, where $f'_{(\alpha,\beta)}$ is also a balanced form.*

**Definition 3.** *For a given balanced form $f(\alpha, \beta; x, y)$, an $(\alpha, \beta)$-RPC applied form $\hat{f}(\alpha, \beta; \gamma)$ is defined as follows:*

$$\hat{f}(\alpha, \beta; \gamma) := \sum_{i=n}^{1} \gamma^{e_n - e_i} \cdot (h_i(x, y)\alpha^{e_i} + g_i(x, y)\beta^{e_i}) + \gamma^{e_n} \cdot c_0(x, y).$$

In a similar fashion, we can define the notion of an $(x, y)$-*RPC applied form* $\hat{f}(x, y; z)$. From the definitions of the RPC applied form and the index, we can directly obtain the following lemma.

**Lemma 1.** *Let $f(\alpha, \beta; x, y)$ be a balanced form. Suppose that $I$ and $e$ are the index and degree of $f_{(\alpha,\beta)}$ respectively and $\hat{f}(\alpha, \beta; \gamma)$ is an $(\alpha, \beta)$-RPC applied form. If $\alpha, \beta, \gamma, x$, and $y$ are regarded as elements in $\mathbb{F}_q^*$ then additionally $I$ field multiplications are required for the computation of $\hat{f}(\alpha, \beta; \gamma)$, compared to $f_{(\alpha,\beta)}$ when ignoring the computations of $\gamma^*$'s.*

Suppose that an extension field $\mathbb{F}_{q^k}$ over $\mathbb{F}_q$ is represented by a polynomial basis $\{1, t, \ldots, t^{k-1}\}$. Now let us consider a polynomial basis equation $F$ on the extension field $\mathbb{F}_{q^k}$ such that

$$F = f_0 + f_1 t + f_2 t^2 + \cdots + f_{k-1} t^{k-1}, \tag{2}$$

where $f_i(\alpha, \beta; x, y)$ is a balanced form over $\mathbb{F}_q$ for each $i = 0, \ldots, k-1$. Then $F$ is called a *balanced form over the extension field* $\mathbb{F}_{q^k}$. Let $I_{i(\alpha,\beta)}$ and $I_{i(x,y)}$ be the indices of $f_{i(\alpha,\beta)}$ and $f_{i(x,y)}$ for each $i = 0, \ldots, k-1$ respectively, and let $e_{i(\alpha,\beta)}$ be the degree of $f_{i(\alpha,\beta)}$ for $i = 0, \ldots, k-1$. For convenience, we present several definitions and notations as follows;

1. $F_{(\alpha,\beta)}$ (resp. $F_{(x,y)}$) denotes a rearranged equation of $F$ with $f_{i(\alpha,\beta)}$ (resp. $f_{i(x,y)}$) for $i = 0, \ldots, k-1$.
2. $\sum_{i=0}^{k-1} I_{i(\alpha,\beta)}$(resp. $\sum_{i=0}^{k-1} I_{i(x,y)}$) is called an $(\alpha, \beta)$(resp. $(x, y)$)-*total index* of $F$, and it is denoted by $I_{(\alpha,\beta)}$(resp. $I_{(x,y)}$).
3. $max\{e_{i(\alpha,\beta)} | i = 0, \ldots, k-1\}$ is called an $(\alpha, \beta)$-*maximum degree* of $F$, and it is denoted by $e_{(\alpha,\beta)}$. In a similar manner, we define an $(x, y)$-maximum degree of $F$, $e_{(x,y)}$.
4. $D_{(\alpha,\beta)}$ is defined as a number of elements of $\{e_{i(\alpha,\beta)} | 0 \neq e_{i(\alpha,\beta)} \neq e_{(\alpha,\beta)}$ for $i = 0, \ldots, k-1\}$, and the notation of $D_{(x,y)}$ has the similar meaning with respect to $x, y$.
5. $C_{(\alpha,\beta)}$(resp. $C_{(x,y)}$) denotes a number of field multiplications for efficiently computing $f_{i(\alpha,\beta)}$(resp. $f_{i(x,y)}$) for all $i = 0, \ldots, k-1$.

**Definition 4.** *An $(\alpha, \beta)$-RPC applied form of $F_{(\alpha,\beta)}$, denoted by $\hat{F}_{(\alpha,\beta)}$, is defined as follows:*

$$\hat{F}_{(\alpha,\beta)} := \sum_{i=0}^{k-1} \gamma^{e_{(\alpha,\beta)} - e_{i(\alpha,\beta)}} \cdot \hat{f}_i(\alpha, \beta; \gamma) t^i.$$

*Similarly, an $(x, y)$-RPC applied form of $F_{(x,y)}$, $\hat{F}_{(x,y)}$, is defined as the same way.*

From Lemma 1, we can prove the following theorem for a computation cost of the RPC applied form of $F$.

**Theorem 1.** *Let $F$ be a balanced form over $\mathbb{F}_{q^k}$ as shown in (2), and $\hat{F}_{(\alpha,\beta)}$ be the $(\alpha,\beta)$-RPC applied form. Suppose that for each $i \neq j \in \{0,\ldots,k-1\}$, $f_{i_{(\alpha,\beta)}}$ and $f_{j_{(\alpha,\beta)}}$ have no same term. If $\alpha,\beta,\gamma,x$, and $y$ are regarded as the elements in $\mathbb{F}_q^*$ then for computing $\hat{F}_{(\alpha,\beta)}$ the required number of field multiplications is as follows:*

$$I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}$$

*when ignoring field multiplications for computing $\gamma^*$'s.*

*Proof.* By the lemma 1, additionally $I_{i_{(\alpha,\beta)}}$ field multiplications are required for computing $\hat{f}_i(\alpha,\beta;\gamma)$ compared to $f_{i_{(\alpha,\beta)}}$ for each $i = 0,\ldots,k-1$, and the number of $\gamma^* \cdot \hat{f}_i(\alpha,\beta;\gamma)$'s is exactly equal to $D_{(\alpha,\beta)}$. Therefore $I_{(\alpha,\beta)} + D_{(\alpha,\beta)}$ additional field multiplications are required for the computation of $\hat{F}_{(\alpha,\beta)}$ since $f_{i_{(\alpha,\beta)}}$ and $f_{j_{(\alpha,\beta)}}$ have no same term for each $i \neq j \in \{0,\ldots,k-1\}$. Hence the proof is completed. □

### 4.3 Construction of Efficient RPC based Countermeasure

Suppose that $F$ be an equation over the extension field in the main loop of a pairing computation algorithm over a given finite field, and $P = (\alpha,\beta)$ and $Q = (x,y)$ be input points of the pairing algorithm.

**Lemma 2.** *Assume that $F$ is a balanced form over the extension field. Then the $(\alpha,\beta)$(resp. $(x,y)$)-RPC applied form of $F$ is an equation over the extension field in the main loop of an RPC based countermeasure randomizing $P = (\alpha,\beta)$(resp. $Q = (x,y)$).*

*Proof.* The proof is essentially based on the idea in [2, 3]. Since $(\alpha,\beta)$ is randomized as $(\alpha,\beta,\gamma) \leftarrow (\gamma\alpha,\gamma\beta,\gamma)$ for $\gamma \in \mathbb{F}_q^*$, we apply $\alpha \leftarrow \frac{\alpha}{\gamma}$ and $\beta \leftarrow \frac{\beta}{\gamma}$ on $F$. Then $F$ is modified by $\frac{1}{\gamma^{e(\alpha,\beta)}}\hat{F}_{(\alpha,\beta)}$. But $(\frac{1}{\gamma^{e(\alpha,\beta)}})^{q-1} = 1$ and the final exponent of the pairing has $(q-1)$ as its factor. Therefore, $\frac{1}{\gamma^{e(\alpha,\beta)}}$ can be ignored on the computation of pairing. □

Lemma 2 and Theorem 1 straightforwardly give us the following corollary on the efficiency of the RPC based countermeasure of the pairing algorithm.

**Corollary 1.** *Let $P = (\alpha,\beta)$ and $Q = (x,y)$ be input points of the pairing algorithm and $F$ be an equation in the main loop of the pairing algorithm. Suppose that $F$ is a balanced form over the extension field as shown in (2), and $f_{i_{(\alpha,\beta)}}$ and $f_{j_{(\alpha,\beta)}}$ have no same term for each $i \neq j \in \{0,\ldots,k-1\}$. Then*

$$I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}$$

*field multiplications are required for computing $\hat{F}_{(\alpha,\beta)}$ when ignoring field multiplications for computing $\gamma^*$'s.*

Let $F$ be a balanced form which is the extension field equation in the main loop of a given pairing algorithm. Then from Corollary 1, $I_{(\alpha,\beta)}+D_{(\alpha,\beta)}+C_{(\alpha,\beta)}$ is a good tool to be able to measure the efficiency of the RPC based countermeasure of the algorithm, since the main computation cost of the pairing algorithm is caused by the field multiplications for computing $F$. If we select one randomizing point between two input points $P = (\alpha, \beta)$ and $Q = (x, y)$, say $P$, then $I_{(\alpha,\beta)}$ and $D_{(\alpha,\beta)}$ values are fixed, but there might be a chance to reduce the value $C_{(\alpha,\beta)}$ because we can modify the coefficient polynomials of $f_{i(\alpha,\beta)}$ for each $i = 0, \ldots, k-1$. Hence, we can propose the following method constructing an efficient and secure pairing algorithm from a given pairing algorithm:

| **Construction of an efficient countermeasure against DPA** |
|---|
| **Step 1.** Determine which method is more efficient between the RPC methods by the point $P$ and $Q$ respectively by investigating $(\_,\_)$-total index, $D_{(\_,\_)}$, and $C_{(\_,\_)}$ of the equation $F$ in the main loop of the algorithm by the corollary 1. |
| **Step 2.** Assume that the method by the point $P$ is selected in the first step. Then modify each $f_{i(\alpha,\beta)}$ for $i = 0, \ldots, k-1$ to reduce the value $C_{(\alpha,\beta)}$ if it is possible. |
| **Step 3.** Apply RPC method randomizing the point $P$ on this modified algorithm |

In the above construction, if we obtain a new modified pairing algorithm of a given pairing algorithm from the step 1 and 2, the modified algorithm can be called as an *RPC-friendly pairing algorithm*, because the modified one leads an efficient RPC based countermeasure against DPA attack. Practical application and examples of our construction method are examined further in the next section.

## 5 Application to Existing Pairing Algorithms

### 5.1 Efficiency of RPC methods randomizing two input points respectively

In this section, we give two examples on the efficiency of the RPC based countermeasures. Firstly we investigate the RPC method on the $\eta_T$ pairing algorithm [1] on supersingular curves in characteristic two, and secondly we examine the RPC method on the Tate pairing algorithm [7] in characteristic three.

Algorithm 1 describes Barreto *et al.*'s $\eta_T$ pairing algorithm over binary fields [1](for details of the closed formula, see Appendix A). $\epsilon_i$ in the step 2 and 5 of Algorithm 1 is defined as in (8) of Appendix A. In [15], Kim *et al.* randomized the input point $Q$ after the comparison of efficiency between RPC based countermeasures by two input points $P$ and $Q$ respectively. But now we can easily check the efficiency of the RPC methods randomizing $P$ and $Q$, respectively by investigating $I_{(\alpha,\beta)}+D_{(\alpha,\beta)}+C_{(\alpha,\beta)}$ and $I_{(x,y)}+D_{(x,y)}+C_{(x,y)}$ of the corollary 1.

**Algorithm 1** $\eta_T$ pairing algorithm on the curve $E_b : Y^2 + Y = X^3 + X + b$ over $\mathbb{F}_{2^m}$ where $b \in \{0, 1\}$ and $m$ odd. [1]

**Input:** $P = (\alpha, \beta)$ and $Q = (x, y)$.
**Output:** $\eta_T(P, \psi(Q))$.
1: $w \leftarrow \alpha + \frac{(m-1)}{2}$
2: $f \leftarrow w \cdot (x + \alpha + 1) + y + (\beta + b + \epsilon_{(m+1)/2}) + (w + x)s + t$
3: **for** $i = 0$ to $(m-1)/2$ **do**
4: $\quad w \leftarrow \alpha + \frac{(m+1)}{2}$, $\alpha \leftarrow \sqrt{\alpha}$, $\beta \leftarrow \sqrt{\beta}$
5: $\quad g \leftarrow w \cdot (\alpha + x + \frac{(m+1)}{2}) + y + (\beta + (1 - \frac{(m+1)}{2})\alpha + \epsilon_{(m-1)/2}) + (w + x)s + t$
6: $\quad f \leftarrow f \cdot g$
7: $\quad$ **if** $i < (m-1)/2$ **then**
8: $\quad\quad x \leftarrow x^2$, $y \leftarrow y^2$
9: $\quad$ **end if**
10: **end for**
11: **return** $f^W = f^{(2^{2m}-1)(2^m+1-\epsilon 2^{(m+1)/2})}$

The extension field equation $F$ in the step 5 of the Algorithm 1 can be exactly described as follows:

$$F := f_0 + f_1 s + t, \text{ where } f_0 = \left(\alpha + \frac{(m+1)}{2}\right)\left(\sqrt{\alpha} + x + \frac{(m+1)}{2}\right) + y + \sqrt{\beta} + \left(1 - \frac{(m+1)}{2}\right)\sqrt{\alpha} + \epsilon_{(m-1)/2} \text{ and } f_1 = \alpha + \frac{(m+1)}{2} + x. \tag{3}$$

*Example 1.* From the equation (3),

$$f_{0(\alpha,\beta)} = \alpha^{3/2} + \left(x + \frac{(m+1)}{2}\right) \cdot \alpha + \left(\alpha^{1/2} + \beta^{1/2}\right) + \left(\frac{(m+1)}{2}(x + \frac{(m+1)}{2}\right) + y + \epsilon_{(m-1)/2}\Big), \; f_{1(\alpha,\beta)} = \alpha + \left(x + \frac{(m+1)}{2}\right), \text{ and}$$

$$f_{0(x,y)} = w(\alpha, \beta) \cdot x + y + c(\alpha, \beta), \; f_{1(x,y)} = x + w(\alpha, \beta), \text{ where} \tag{4}$$

$$w(\alpha, \beta) = \left(\alpha + \frac{(m+1)}{2}\right) \text{ and} \tag{5}$$

$$c(\alpha, \beta) = \alpha^{3/2} + \frac{(m+1)}{2}\alpha + \alpha^{1/2} + \beta^{1/2} + \frac{(m+1)}{2} + \epsilon_{(m-1)/2}. \tag{6}$$

Then the $(\alpha, \beta)$(resp. $(x, y)$)-total index of $F$ is 4(resp. 2), $D_{(\alpha,\beta)} = 1$ and $D_{(x,y)} = 0$. $f_{1(\alpha,\beta)}$ and $f_{1(x,y)}$ do not require any field multiplication. $f_{0(\alpha,\beta)}$ needs two field multiplications(for computing $\alpha^{3/2} = \alpha \cdot \alpha^{1/2}$ and $\left(x + \frac{(m+1)}{2}\right) \cdot \alpha$), and two field multiplications are required for $f_{0(x,y)}$(for computing $w(\alpha, \beta) \cdot x$ and $\alpha^{3/2}$). Therefore $I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)} = 4 + 1 + 2 = 7$ and $I_{(x,y)} + D_{(x,y)} + C_{(x,y)} = 2 + 0 + 2 = 4$. Since the base field squaring are relatively inexpensive [11] and the method in [9] for computing square roots is as fast as squaring, the field multiplication cost is sufficient to compare the efficiencies of these two

RPC methods. Hence the RPC method by the point $Q$ is more efficient than the method by $P$.

*Example 2.* In Duursma-Lee's Tate pairing algorithm over fields with characteristic three [7, 18], the equation in the main loop is as follows:

$$F := f_0 - f_1\sigma - f_2\rho - \rho^2, \text{ where } f_0 = -f_2^2,\ f_1 = \beta^3 y\ f_2 = \alpha^3 + x + b,\ b = \pm 1$$

for given input points $P = (\alpha, \beta)$ and $Q = (x, y)$. Then the indices of $f_{0(\alpha,\beta)}$ and $f_{2(\alpha,\beta)}$(resp. $f_{0(x,y)}$ and $f_{2(x,y)}$) are both 1 since $f_0 = -f_2^2$. Therefore $I_{(\alpha,\beta)} = 1 + 0 + 1 + 0 = I_{(x,y)}$. Furthermore, $D_{(\alpha,\beta)} = 2 = D_{(x,y)}$ and $C_{(\alpha,\beta)} = 2 = C_{(x,y)}$. Since the constant term $(x + b)$(resp. $(\alpha^3 + b)$) of $f_{0(\alpha,\beta)}$(resp. $f_{0(x,y)}$) is repeated at $f_{2(\alpha,\beta)}$(resp. $f_{2(x,y)}$), we can reduce the field multiplications by one for computing $F_{(\alpha,\beta)}$(resp. $F_{(x,y)}$). Therefore $(I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}) - 1 = 5$(resp. $(I_{(x,y)} + D_{(x,y)} + C_{(x,y)}) - 1 = 5$) field multiplications are required for computing $\hat{F}_{(\alpha,\beta)}$(resp. $\hat{F}_{(x,y)}$). Therefore the RPC methods by the point $P$ and $Q$ have the same field multiplication cost when ignoring the cubing and the cubic root computations. But since computing the cubic root is generally more expensive than the cubing computation, we can conclude that the RPC method by the point $P$ is more efficient than the method by $Q$.

### 5.2 RPC-Friendly $\eta_T$ Pairing Algorithm over Binary Fields

In this section, we induce an RPC-friendly algorithm on the $\eta_T$ pairing over binary fields using the step 1 and 2 of our construction method.

1. Since the RPC-method by the point $Q = (x, y)$ is more efficient than the method by the point $P$ by the Example 1, firstly we rearrange the equation (3) in the main loop of the Algorithm 1 with $(x, y)$-variable as in the equations (4), (5), and (6).
2. Secondly, we modify the coefficient polynomial $c(\alpha, \beta)$ in (6) as follows:

$$\begin{aligned} c(\alpha,\beta) &= \left(\alpha^3 + \alpha\right)^{1/2} + \frac{(m+1)}{2}\alpha + \beta^{1/2} + \frac{(m+1)}{2} + \epsilon_{(m-1)/2} \\ &\quad \text{by the Weierstrass equation of } E_b : Y^2 + Y = X^3 + X + b, \\ &= \left(\beta^2 + \beta + b\right)^{1/2} + \frac{(m+1)}{2}\alpha + \beta^{1/2} + \frac{(m+1)}{2} + \epsilon_{(m-1)/2} \\ &= \frac{(m+1)}{2}\alpha + \beta + \frac{(m+1)}{2} + b + \epsilon_{(m-1)/2} \\ &= \frac{(m+1)}{2}w(\alpha,\beta) + \beta + b + \epsilon_{(m-1)/2}, \end{aligned} \tag{7}$$

   where $w(\alpha, \beta)$ is defined as (5). Consequently, the modified $c(\alpha, \beta)$ does not require any field multiplication. Therefore, $C_{(x,y)}$ is reduced by one.

Note that more explicitly the equation (7) shows that $g_{2^{-j}P'}(\psi(Q))^{2^j} =$

$$w_j x^{(j)} + y^{(j)} + \left(\frac{(m+1)}{2}w_j + \beta^{(-j)} + b + \epsilon_{(m-1)/2}\right) + (w_j + x^{(j)})s + t,$$

where $w_j = (\alpha^{(-j)} + \frac{(m+1)}{2})$(for details of the notations, see Appendix A).

**Algorithm 2** RPC-friendly algorithm of $\eta_T$ Pairing on the curve $E_b : Y^2 + Y = X^3 + X + b$ over $\mathbb{F}_{2^m}$ where $b \in \{0, 1\}$ and $m$ odd.

**Input:** $P = (\alpha, \beta)$ and $Q = (x, y)$.
**Output:** $\eta_T(P, \psi(Q))$.
1: $w \leftarrow \alpha + \frac{(m-1)}{2}$
2: $f \leftarrow w \cdot (x + \alpha + 1) + y + (\beta + b + \epsilon_{(m+1)/2}) + (w + x)s + t$
3: **for** $i = 0$ to $(m-1)/2$ **do**
4: $\quad w \leftarrow \alpha + \frac{(m+1)}{2}$
5: $\quad g \leftarrow w \cdot x + y + (\frac{(m+1)}{2}w + \beta + b + \epsilon_{(m-1)/2}) + (w + x)s + t$
6: $\quad f \leftarrow f \cdot g$
7: $\quad$ **if** $i < (m-1)/2$ **then**
8: $\quad\quad \alpha \leftarrow \sqrt{\alpha}$, $\beta \leftarrow \sqrt{\beta}$, $x \leftarrow x^2$, $y \leftarrow y^2$
9: $\quad$ **end if**
10: **end for**
11: **return** $f^W = f^{(2^{2m}-1)(2^m+1-\epsilon 2^{(m+1)/2})}$

Our new RPC-friendly $\eta_T$ pairing algorithm is as shown in Algorithm 2. Algorithm 2 just reduces the computational cost by two square root computations on the base field $\mathbb{F}_q$, compared with the Algorithm 1(*i.e.* Barreto *et al.* algorithm), and so this modification is only negligible amount of total computational cost of $\eta_T$ pairing. That is, the field multiplication costs of the Algorithm 1 and the Algorithm 2 are the same. Nevertheless, in the RPC based countermeasure point of view, this small modification induces a meaningful difference.

In Algorithm 2, $(x, y)$-total index of the equation in the main loop is 2, $D_{(x,y)} = 0$, and $C_{(x,y)} = 1$. Therefore $I_{(x,y)} + D_{(x,y)} + C_{(x,y)} = 3$ field multiplications are required for computing the equation in the main loop of the RPC-based countermeasure on the Algorithm 2 by the corollary 1. But 4 field multiplications are required for computing the equation in the main loop of the RPC method(*i.e.* Kim *et al.* algorithm [15]) on the original Alogorithm 1(*i.e.* Barreto *et al.* algorithm [1])(see Example 1).

Algorithm 3 describes the RPC based countermeasure applied on our RPC-friendly algorithm. The total field multiplication cost of Algorithm 3 is $6(m + 1)M + 5M$ since the total cost of Kim *et al.* algorithm is $6.5(m + 1)M + 5M$ [15], where $M$ means one base field multiplication cost.

Since the total cost of Barreto *et al.* algorithm and our RPC-friendly algorithm is $3.5(m+1)M+1M$ [15], the additional field multiplication cost of Algorithm 3(resp. Kim *et al.* algorithm) is $2.5(m+1)M+4M$(resp. $3(m+1)M+4M$). Consequently, our Algorithm 3 reduces the additional cost by 17% for $m = 239$, compared with Kim *et al.* algorithm [15].

**Algorithm 3** Efficient and Secure $\eta_T$ Pairing Algorithm on the curve $E_b : Y^2 + Y = X^3 + X + b$ over $\mathbb{F}_{2^m}$ where $b \in \{0, 1\}$ and $m$ odd.

**Input:** $P = (\alpha, \beta)$ and $Q = (x, y)$.
**Output:** $\eta_T(P, \psi(Q))$.

Choose $\bar{z} \in \mathbb{F}_q^*$ at random
$\bar{x} \leftarrow \bar{z}x$, $\bar{y} \leftarrow \bar{z}y$
$w \leftarrow \alpha + \frac{(m-1)}{2}$
$f \leftarrow w \cdot (\bar{x} + \bar{z} \cdot (\alpha + 1)) + \bar{y} + \bar{z} \cdot (\beta + b + \epsilon_{(m+1)/2}) + (\bar{z} \cdot w + \bar{x})s + \bar{z}t$
**for** $i = 0$ to $(m-1)/2$ **do**
  $w \leftarrow \alpha + \frac{(m+1)}{2}$
  $g \leftarrow w \cdot \bar{x} + \bar{y} + \bar{z} \cdot (\frac{(m+1)}{2}w + \beta + b + \epsilon_{(m-1)/2}) + (\bar{z} \cdot w + \bar{x})s + \bar{z}t$
  $f \leftarrow f \cdot g$
  **if** $i < (m-1)/2$ **then**
    $\alpha \leftarrow \sqrt{\alpha}$, $\beta \leftarrow \sqrt{\beta}$, $\bar{x} \leftarrow \bar{x}^2$, $\bar{y} \leftarrow \bar{y}^2$, $\bar{z} \leftarrow \bar{z}^2$
  **end if**
**end for**
**return** $f^W = f^{(2^{2m}-1)(2^m+1-\epsilon 2^{(m+1)/2})}$

## 6 Conclusion

In this study, we have performed a measurement of the pairing computing algorithm in order to estimate the efficiency of an RPC-based countermeasure against SCAs. We have been able to construct a method to yield an efficient countermeasure of the pairing algorithm against SCAs. Using this method, we have presented an RPC-friendly $\eta_T$ pairing algorithm over binary fields from the original Barreto *et al.*'s algorithm. The proposed RPC-friendly $\eta_T$ pairing algorithm reduces the computation cost by two square root computations and has only a slight advantage in efficiency. However, if we apply the RPC method to this algorithm as protection against DPA attacks, then this countermeasure reduces the additional computation cost by 17%, compared with that in the case of application of the RPC method [15] to Barreto *et al.*'s algorithm, which is the most efficient existing countermeasure. This implies that a small modification of the original algorithm might have a significant effect on the efficiency of DPA countermeasures.

## 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, "On the selection of pairing-friendly groups," *SAC 2003*, LNCS 3006, pp.17-25, 2004.
4. D. Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing," *SIAM J. of Computing*, Vol.32, No.3, pp.586-615, 2003.

5. D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," *Journal of Cryptology*, Vol.17, No.4, pp.297-319, 2004.
6. J.C. Cha and J.H. Cheon, "An Indentity-Based Signature from Gap Diffie-Hellman Groups," *PKC 2003*, LNCS 2567, pp.18-30, 2003.
7. 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.
8. G. Frey and H.G. Rück, "A remark concening $m$-divisibility and the discrete logarithm in the divisor class group of curves," *Math. Comput.*, Vol.62, pp.865-874, 1994.
9. 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.
10. S.D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," *ANTS V*, LNCS 2369, pp.324-337, 2002.
11. 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.
12. F. Hess, "Exponent group signature schemes and efficient identity based signature schems based on pairing," *SAC 2002*, LNCS 2595, pp.310-324, 2002.
13. F. Hess, N. Smart, and F. Vercauteren, "The eta pairing revisited," IEEE Trans. Inf. Theory. 52 no. 10 pp.4595-4602, 2006.
14. A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman," *Journal of Cryptology*, Vol.17, No.4, pp.263-276, 2004.
15. T. H. Kim, T. Takagi, D.-G. Han, H. W. Kim and J. Lim, "Side Channel Attacks and Countermeasures on Pairing Based Cyrptosystems over Binary Fields," *CANS 2006*, LNCS 4301, pp.168-181, 2006.
16. P. Kocher,"Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *CRYPTO 1996*, LNCS 1109, pp.104-113, 1996.
17. C. Kocher, J. Jaffe, B. Jun,"Differential Power Analysis," *CRYPTO 1999*, LNCS 1666, pp.388-397, 1999.
18. S. Kwon, "Efficient Tate Pairing Computation for Elliptic Curves over Binary Fields,"*ACISP 2005*, LNCS 3574, pp.134-145, 2005.
19. A.K. Lenstra and E.R. Verheul, "The XTR public key system," *Advances in Cryptology - CRYPTO '00*, LNCS 1880, pp.1-19, 2000.
20. A. Menezes, "Elliptic Curve Public Key Cryptosystems," Kluwer Academic Publishers, 1993.
21. A. Menezes, T. Okamoto, and S. Vanstone "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field," *IEEE Trans. Inform. Theory*, Vol.39, No.5, pp.1639-1646, 1993.
22. V. Miller, "Short Programs for Functions on Curves," unpublished manuscript, 1986.
23. D. Page and F. Vercauteren, "Fault and Side-Channel Attacks on Pairing Based Cryptography," Cryptology ePrint Archive, Report 2004/283, 2005. `http://eprint.iacr.org/2004/283`.
24. D. Page and F. Vercauteren, "A Fault Attack on Pairing Based Cryptography," To appear in IEEE Transactions on Computers 2006.
25. K.G. Paterson, "ID-based signature from pairings on elliptic curves," *Electronics Letters*, Vol.38, No.18, pp.1025-1026, 2002.
26. 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`.

27. M. Scott, "Computing the Tate Pairing," *CT-RSA 2005*, LNCS 3376, pp.293-304, 2005.
28. N.P. Smart, "An identity based authentication key agreement protocol based on pairing," *Electronics Letters*, Vol.38, No.13, pp.630-632, 2002.
29. 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 Pairing Computation over Binary Fields

In this appendix, we briefly review the closed formula of the Tate and $\eta_T$ pairing over binary fields in [1, 18]. We consider the elliptic curves over binary fields $\mathbb{F}_q$, where $q = 2^m$ and $m$ is odd, as follows;

$$E_b : Y^2 + Y = X^3 + X + b, \text{ where } b \in \{0, 1\}.$$

Then $E_b$ has the embedding degree $k = 4$[20, 1, 18] and $\#E_b(\mathbb{F}_q) = 2^m + 1 + \epsilon 2^{\frac{m+1}{2}}$, where

$$\epsilon = \begin{cases} -1, & \text{if } (m = 1, 7 \pmod 8 \text{ and } b = 1) \text{ or } (m = 3, 5 \pmod 8 \text{ and } b = 0), \\ 1, & \text{otherwise.} \end{cases}$$

In this elliptic curve $E_b$, the extension field $\mathbb{F}_{q^4}$ is represented by the basis $\{1, s, t, st\}$ such that $s^2 + s + 1 = 0$ and $t^2 + t + s = 0$, and the distortion map is $\psi(x, y) = (x + s^2, y + xs + t)$. Furthermore, in this setting, Barreto et al. [1] showed that $T$ value of the $\eta_T$ pairing is $2^{(m+1)/2} + \epsilon$.

It can be directly induced that for a given $P = (\alpha, \beta)$,

$$2^i P = (\alpha_i^{(2i)}, \beta_i^{(2i)}), \text{ where } (x_i, y_i) = \phi^i(x, y), \ \phi(x, y) = (x + 1, y + x + 1).$$

In the above equation, $\alpha^{(j)}$(resp. $\beta^{(j)}$) is defined as $\alpha^{(j)} = \alpha^{2^j}$(resp. $\beta^{(j)} = \beta^{2^j}$)(for more details, see [1, 18]). Then

$$\phi^i(x, y) = (x + i, y + ix + \epsilon_i), \text{ where } \epsilon_i = \begin{cases} 0 & \text{if } 0, 1 \mod 4, \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

In [18], Kwon gave a closed formula of the Tate pairing, and Barreto *et al.* [1] independently found a closed formula of the $\eta_T$ pairing on the elliptic curve $E_b$. The following theorem is a summary of these results.

**Theorem 2 ([1, 18]).** *For given $P = (\alpha, \beta), Q = (x, y)$ in $E_b(\mathbb{F}_q)$,*

1. *The Tate pairing $\tau_l(P, \psi(Q)) =$*

$$\left( \prod_{i=0}^{m-1} g_{2^i P}(\psi(Q))^{2^{2m-i}} \right)^{2^{2m}-1},$$

*where $g_R(X, Y)$ is an equation of the tangent line at $R$.*

*2. The $\eta_T$ pairing* $\eta_T(P, \psi(Q)) =$

$$\left( \left( \prod_{j=0}^{(m-1)/2} g_{2^{-j}P'}(\psi(Q))^{2^j} \right) \cdot \ell(\psi(Q)) \right)^W,$$

*where* $P' = 2^{(m-1)/2}P$, $\ell(X, Y)$ *is an equation of line passing* $2^{m+1/2}P$ *and* $\epsilon P$, *and* $W = q^k - 1/N = (2^{2m} - 1)(2^m + 1 - \epsilon 2^{m+1/2})$, $N = \#E_b(\mathbb{F}_q)$.

*Furthermore,* $g_{2^iP}(\psi(Q)) =$

$$(\alpha_i^{(2i+1)} + 1)(x + 1) + y + \beta_i^{(2i+1)} + b + (\alpha_i^{(2i+1)} + 1 + x)s + t \tag{9}$$

From the equations (8) and (9), it can be straightforwardly proved [18] that

$$\begin{aligned} g_{2^iP}(\psi(Q))^{2^{2m-i}} =& (\alpha^{(i+1)} + 1) \cdot x^{(-i)} + y^{(-i)} + (\alpha^{(i+1)} + \beta^{(i+1)} + b) \\ &+ ((\alpha^{(i+1)} + 1) + x^{(-i)})s + t, \end{aligned}$$

and Barreto *et al.* [1] computed that

$$\begin{aligned} g_{2^{-j}P'}(\psi Q)^{2^j} =& w_j(\alpha^{(-1-j)} + x^{(j)} + \frac{(m+1)}{2}) + y^{(j)} + \beta^{(-1-j)} \\ &+ (1 - \frac{(m+1)}{2})\alpha^{(-1-j)} + \epsilon_{(m-1)/2} + (w_j + x^{(j)})s + t, \qquad (10) \\ \ell(\psi(Q)) =& (\alpha + (m-1)/2) \cdot (\alpha + x + 1) + y + (\beta + b + \epsilon_{(m+1)/2}) \\ &+ ((\alpha + (m-1)/2) + x)s + t, \end{aligned}$$

where $w_j = (\alpha^{(-j)} + \frac{(m+1)}{2})$.