An Efficient Residue Group Multiplication for The η_T Pairing Over $\mathbb{F}_{3^m}$

Yuta Sasaki, Satsuki Nishina, Masaaki Shirase, and Tsuyoshi Takagi

Future University Hakodate

Abstract. When we implement the η_T pairing, which is one of the fastest pairings, we need multiplications in a base field $\mathbb{F}_{3^m}$ and in a group G. We have previously regarded elements in G as those in $\mathbb{F}_{3^{6m}}$ to implement the η_T pairing. Gorla et al. proposed a multiplication algorithm in $\mathbb{F}_{3^{6m}}$ that takes 5 multiplications in $\mathbb{F}_{3^{2m}}$, namely 15 multiplications in $\mathbb{F}_{3^m}$. This algorithm then reaches the theoretical lower bound of the number of multiplications. On the other hand, we may also regard elements in G as those in the residue group $\mathbb{F}_{3^{6m}}^* / \mathbb{F}_{3^m}^*$ in which βa is equivalent to a for $a \in \mathbb{F}_{3^{6m}}^*$ and $\beta \in \mathbb{F}_{3^m}^*$. This paper proposes an algorithm for computing a multiplication in the residue group. Its cost is asymptotically 12 multiplications in $\mathbb{F}_{3^m}$ as $m \to \infty$, which reaches beyond the lower bound the algorithm of Gorla et al. reaches. The proposed algorithm is especially effective when multiplication in the finite field is implemented using a basic method such as shift-and-add.

Key words: Finite field multiplication, pairing, residue group, Vandermonde matrix.

1 Introduction

Most public key cryptosystems (PKCs) are mainly computed using multiplications in finite fields, thus polynomial multiplications are important to efficiently implement PKCs because elements in the finite fields are represented as polynomials. The algorithms that most efficiently compute polynomial multiplications are those derived by Karatsuba [12], Toom-Cook [17, 10, 4], Cantor [9], and Schönhage [15]. Karatsuba's algorithm is suitable for polynomial multiplications of small and medium degrees, Toom-Cook's algorithm is suitable for those of medium degrees, and Cantor's and Schönhage's algorithms are suitable for those of large degrees. Brent et al. [8] inclusively improved these algorithms for $\mathbb{F}_2[x]$.

Recently, pairing based cryptosystems (PBCs) such as an identity-based encryption [6], an efficient broadcast encryption [7], and a keyword searchable encryption [5] have been attracting attention. For PBCs, we need multiplications in a base field $\mathbb{F}_q$ and in a group G. We have regarded elements in G as those in $\mathbb{F}_{q^k}$ to implement pairings, where k is an integer called the embedding degree. PBCs are practical when k is small. Thus multiplications in $\mathbb{F}_{q^k}$ are generally implemented using Karatsuba's algorithm.

The η_T pairing proposed by Barreto et al. [1] is one of the fastest pairings. It is defined over $\mathbb{F}_{3^m}$ or $\mathbb{F}_{2^m}$, and the embedding degrees become 6 or 4, respectively, where m has to be a prime number for PBC security. This paper focuses on multiplications on $\mathbb{F}_{3^{6m}}$ to efficiently implement the η_T pairing. Arithmetic in $\mathbb{F}_{3^{6m}}$ is generally implemented using a tower of extensions $\mathbb{F}_{3^m} \subset \mathbb{F}_{3^{2m}} \subset \mathbb{F}_{3^{6m}}$ that Kerins et al. [13], Gorla et al. [11] and Beuchat et al. [3] used.

Using Karatsuba's algorithm, a multiplication in $\mathbb{F}_{3^{2m}}$ is computed by 3 multiplications and a multiplication in $\mathbb{F}_{3^{6m}}$ is computed by 6 multiplications. Then 18 multiplications in $\mathbb{F}_{3^m}$ are needed. Additionally, a polynomial multiplication of degree t needs at least $2t + 1$ multiplications according to the theory of multiplicative complexity (see Lempel et al. [14] and Winograd [18]). Then a multiplication in $\mathbb{F}_{3^{2m}}$, which needs 3 multiplications in $\mathbb{F}_{3^m}$, reaches the lower bound because elements in $\mathbb{F}_{3^{2m}}$ are represented as the polynomials degree of 1. On the other hand, a multiplication in $\mathbb{F}_{3^{6m}}$, which needs 6 multiplications in $\mathbb{F}_{3^{2m}}$, does not yet reach the lower bound, which is 5. Gorla et al. proposed a multiplication algorithm for $\mathbb{F}_{3^{6m}}$ that takes 5 multiplications in $\mathbb{F}_{3^{2m}}$, namely 15 multiplications in $\mathbb{F}_{3^m}$, using the 4×4 Vandermende matrix, all the coefficients of which are the fourth roots of unity. Thus this algorithm reaches the lower bound.

In this paper, we regard elements in G as those in the residue group $\mathcal{G} = \mathbb{F}_{3^{6m}}^* / \mathbb{F}_{3^m}^*$. In $\mathcal{G}$, βa is equivalent to a for $a \in \mathbb{F}_{3^{6m}}^*$ and $\beta \in \mathbb{F}_{3^m}^*$. The aim of this paper is to propose a *residue group multiplication* (RGM) algorithm in $\mathcal{G}$, which is a modification of an algorithm of Shirase et al. [16] to the case of characteristic 3. The cost of the proposed RGM algorithm is asymptotically 12 multiplications in $\mathbb{F}_{3^m}$ as $m \to \infty$, which reaches beyond the lower bound the algorithm of Gorla et al. reaches. In the proposed RGM algorithm, $\mathbb{F}_{3^{6m}}$ is directly represented as the sixth extension of $\mathbb{F}_{3^m}$ unlike current implementation as done by Kerins et al. [13], Gorla et al. [11], and Beuchat et al [3]. Consequently we can use a Vandermonde matrix (8×8) bigger than that used in the algorithm of Gorla et al. (4×4). This bigger Vandermond matrix reduces the cost of the proposed RGM algorithm.

Moreover, we implemented the η_T pairing over $\mathbb{F}_{3^{97}}$, which for security had 1,024-bit RSA on a Core 2 Duo E6320 1.86GHz with 1GB RAM using gcc 3.4.4. Using the algorithm of Gorla et al. and the proposed RGM algorithm, we then compared timings of the η_T pairings. Consequently, the timing of the η_T pairing using the proposed RGM algorithm was almost 5 percent faster than that using the algorithm of Gorla et al.

This paper is organized as follows: In Section 2 we explain the η_T pairing over $\mathbb{F}_{3^m}$. We explain multiplication algorithm in $\mathbb{F}_{3^{6m}}$ in Section 3. In Section 4 we present our proposed RGM algorithm. Lastly, we conclude this paper in Section 5.

2 Implementation of the η_T Pairing over $\mathbb{F}_{3^m}$

In this section, we explain implementations of finite fields $\mathbb{F}_{3^m}$ and $\mathbb{F}_{3^{6m}}$, and the η_T pairing over $\mathbb{F}_{3^m}$.

2.1 Finite Field $\mathbb{F}_{3^m}$ and Extension Field $\mathbb{F}_{3^{6m}}$

.

Let $\mathbb{F}_3 = \{0, 1, 2\}$ be the prime field with characteristic 3. First, we explain how $\mathbb{F}_3$ is represented on computers by following the method of Kerins et al. [13]. An element $a \in \mathbb{F}_3$ is represented by two bits such as $a = (a_{hi}, a_{lo})$ for $a_{hi}, a_{lo} \in \{0, 1\}$, specifically, $(0, 0), (0, 1), (1, 0)$, mean 0, 1, 2, respectively. Note that the negative $-a$ for $a \in \mathbb{F}_3$ is replaced by $2a$, and it is represented by $-a = (a_{lo}, a_{hi})$ for $a = (a_{hi}, a_{lo})$.

Let $\mathbb{F}_3[x]$ be a set of polynomials with coefficients in $\mathbb{F}_3$. Then a finite field $\mathbb{F}_{3^m}$ is represented as

$$\mathbb{F}_{3^m} = \mathbb{F}_3[x] \,/\, f(x),$$

where $f(x)$ is an irreducible polynomial of degree m. Let A be an element in $\mathbb{F}_{3^m}$. A can be represented as the polynomial of degree at most $m-1$ as

$$A = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + ... + a_1 x + a_0.$$

$\mathbb{F}_{3^{6m}}$ is the sixth extension field of $\mathbb{F}_{3^m}$. Let $g(\sigma)$ and $h(\rho)$ be irreducible polynomials with $g(\sigma) = \sigma^2 + 1$ over $\mathbb{F}_{3^m}$ and $h(\rho) = \rho^3 - \rho - 1$ over $\mathbb{F}_{3^{2m}}$. We then follow the tower field representation of Kerins et al. [13],

$$\mathbb{F}_{3^{2m}} = \mathbb{F}_{3^m}[\sigma] \,/\, g(\sigma),$$
$$\mathbb{F}_{3^{6m}} = \mathbb{F}_{3^{2m}}[\rho] \,/\, h(\rho).$$

Let A_0, A_1, A_2 be elements in $\mathbb{F}_{3^{2m}}$ with $A_0 = a_1\sigma + a_0$, $A_1 = a_3\sigma + a_2$ and $A_2 = a_5\sigma + a_4$. Then $A' \in \mathbb{F}_{3^{6m}}$ is represented as

$$A' = A_2\rho^2 + A_1\rho + A_0 = a_5\sigma\rho^2 + a_4\rho^2 + a_3\sigma\rho + a_2\rho + a_1\sigma + a_0.$$

Then a set $\{\sigma\rho^2, \rho^2, \sigma\rho, \rho, \sigma, 1\}$ forms a base of $\mathbb{F}_{3^{6m}}$ over $\mathbb{F}_{3^m}$. We call it $\sigma\rho$ base in this paper. Note that the roots of $\sigma^2 + 1$ are the primitive fourth roots of unity since $\sigma^2 = -1$ and $\sigma \neq \pm 1$.

Let $\mathbb{F}^*_{3^m}$ be the multiplicative group of $\mathbb{F}_{3^m}$, and let $\mathbb{F}^*_{3^{6m}}$ be the multiplicative group of $\mathbb{F}_{3^{6m}}$. That is, $\mathbb{F}^*_{3^m} = \mathbb{F}_{3^m} - \{0\}$ and $\mathbb{F}^*_{3^{6m}} = \mathbb{F}_{3^{6m}} - \{0\}$.

2.2 η_T Pairing over $\mathbb{F}_{3^m}$

Let E be a supersingular curve $E : y^2 = x^3 - x + b, b = \pm 1$ over $\mathbb{F}_{3^m}$. Then the η_T pairing is a bilinear map

$$\eta_T : E(\mathbb{F}_{3^m})[r] \times E(\mathbb{F}_{3^m})[r] \rightarrow \mathbb{F}^*_{3^{6m}} \,/\, (\mathbb{F}^*_{3^{6m}})^r,$$

where r is the largest prime number such that $r \mid \#E(\mathbb{F}_{3^m})$, and 6 is the embedding degree. The η_T pairing satisfies the equation $\eta_T(aP, Q) = \eta_T(P, aQ) = \eta_T(P, Q)^a$ for any integer $a \neq 0$.

We used an algorithm for computing the η_T pairing used in [3], which is efficient due to it not having a cube root operation (Algorithm 1).

Algorithm 1 The η_T pairing algorithm without a cube root operation [3]

INPUT: $P(x_p, y_p), Q(x_q, y_q) \in E(\mathbb{F}_{3^m})[r]$
OUTPUT: $\eta_T(P, Q) \in \mathbb{F}_{3^{6m}}$
1: $y_p \leftarrow -y_p, d \leftarrow 1$
2: $R_0 \leftarrow -y_p(x_p + x_q + 1) + y_q\sigma + y_p\rho$
3: **for** $i \leftarrow 0$ to $(n-1)/2$ **do**
4: $\quad v \leftarrow x_p + x_q + d$
5: $\quad R_1 \leftarrow -v^2 + y_p y_q \sigma - v\rho - \rho^2$
6: $\quad R_0 \leftarrow R_0 R_1$
7: $\quad y_p \leftarrow -y_p$
8: $\quad x_q \leftarrow x_q^9, y_q \leftarrow y_q^9$
9: $\quad d \leftarrow ((d-1) \bmod 3)$
10: $\quad R_0 \leftarrow R_0^3$
11: **end for**
12: **return** R_0

Remark 1.
Elements in $\mathbb{F}_{3^{6m}}^* / (\mathbb{F}_{3^{6m}}^*)^r$ have previously been regarded as those in $\mathbb{F}_{3^{6m}}$ to implement the η_T pairing. However, note that elements in $\mathbb{F}_{3^{6m}}^* / (\mathbb{F}_{3^{6m}}^*)^r$ may be regarded as those in the residue group $\mathcal{G} = \mathbb{F}_{3^{6m}}^* / \mathbb{F}_{3^m}^*$ because $(\mathbb{F}_{3^{6m}}^*)^r$ is a subgroup of $\mathbb{F}_{3^m}^*$. In $\mathcal{G}$, βa is equivalent to a for $a \in \mathbb{F}_{3^{6m}}^*$ and $\beta \in \mathbb{F}_{3^m}^*$.

3 Multiplication Algorithm in $\mathbb{F}_{3^{6m}}$

In this section, we explain Karatsuba's algorithm [12] and the multiplication algorithm by Gorla et al. [11].

3.1 Karatsuba's Algorithm [12]

Karatsuba's algorithm is generally used in a multiplication algorithm in $\mathbb{F}_{3^{6m}}$.

We consider a case in which $\mathbb{F}_{3^{6m}}$ is implemented using a tower of extensions $\mathbb{F}_{3^m} \subset \mathbb{F}_{3^{2m}} \subset \mathbb{F}_{3^{6m}}$ that Kerins et al. [13], Gorla et al. [11], and Beuchat et al. [3] used.

Let $A(\rho), B(\rho)$ be elements in $\mathbb{F}_{3^{6m}}$ with $A(\rho) = a_2\rho^2 + a_1\rho + a_0$ and $B(\rho) = b_2\rho^2 + b_1\rho + b_0$. Multiplication in $\mathbb{F}_{3^{6m}}$ is defined by $A(\rho) \cdot B(\rho) \bmod h(\rho)$. Let $P(\rho) = A(\rho) \cdot B(\rho) \bmod h(\rho)$. Let

$$t_1 = a_2(b_0 + b_2),\ t_2 = a_1(b_1 + b_2),\ t_3 = a_0(b_0 - b_1),$$
$$t_4 = b_2(a_0 - a_1),\ t_5 = b_1(a_0 - a_1 + a_2),\ t_6 = b_0(a_1 - a_2).$$

$P(\rho)$ is computed by Karatsuba's algorithm as follows:

$$P(\rho) = (t_1 + t_2 + t_4)\rho^2 + (t_1 + t_2 + t_5 + t_6)\rho + (t_2 + t_3 + t_5).$$

Thus, multiplication in $\mathbb{F}_{3^{6m}}$ is computed by 6 multiplications in $\mathbb{F}_{3^{2m}}$.

Next, let $A'(\sigma), B'(\sigma)$ be elements in $\mathbb{F}_{3^{2m}}$ with $A'(\sigma) = a'_1\sigma + a'_0$ and $B'(\sigma) = b'_1\sigma + b'_0$. Let $Q(\sigma) = A'(\sigma) \cdot B'(\sigma) \bmod g(\sigma)$. $Q(\sigma)$ is computed by Karatsuba's algorithm by

$$Q(\sigma) = (u_1 + u_3)\sigma + (u_2 + u_3),$$

where $u_1 = a'_1(b'_0 + b'_1), u_2 = a'_0(b'_0 - b'_1), u_3 = b'_1(a'_0 - a'_1)$. Thus, multiplication in $\mathbb{F}_{3^{2m}}$ is computed by 3 multiplications in $\mathbb{F}_{3^m}$. Therefore, multiplication in $\mathbb{F}_{3^{6m}}$ can be obtained by 18 multiplications in $\mathbb{F}_{3^m}$.

3.2 Multiplication Algorithm of Gorla et al. [11]

The algorithm of Gorla et al. [11] can compute a multiplication in $\mathbb{F}_{3^{6m}}$ most efficiently. Indeed it computes a multiplication in $\mathbb{F}_{3^{6m}}$ with 5 multiplications in $\mathbb{F}_{3^{2m}}$, which theoretically reaches the lower bound [14, 18] because a polynomial multiplication of degree m needs at least $2m + 1$ multiplications according to the theory of multiplicative complexity.

The algorithm of Gorla et al. uses the primitive fourth root of unity and the Vandermonde matrix. Let $A(\rho), B(\rho)$ be elements in $\mathbb{F}_{3^{6m}}$ with $A(\rho) = a_2\rho^2 + a_1\rho + a_0$ and $B(\rho) = b_2\rho^2 + b_1\rho + b_0$. Let $C(\rho)$ be a product $A(\rho)$ and $B(\rho)$,

$$C(\rho) = A(\rho) \cdot B(\rho) = c_4\rho^4 + c_3\rho^3 + c_2\rho^2 + c_1\rho + c_0.$$

Note that we refer to σ as the primitive fourth root of unity in Section 2.1 and σ as the generator of the base of $\mathbb{F}_{3^{2m}}$ over $\mathbb{F}_{3^m}$. Let $Y = (1, \sigma^1, \sigma^2, \sigma^3) = (1, \sigma, -1, -\sigma)$, and let V_σ be the Vandermonde matrix for Y as follows:

$$V_\sigma = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \sigma & -1 & -\sigma \\ 1 & -1 & 1 & -1 \\ 1 & -\sigma & -1 & \sigma \end{pmatrix}. \tag{1}$$

Coefficients of $C(\rho)$ satisfy the following matrix.

$$\begin{aligned} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \sigma & -1 & -\sigma \\ 1 & -1 & 1 & -1 \\ 1 & -\sigma & -1 & \sigma \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} &= \begin{pmatrix} c_0 + c_1 + c_2 + c_3 \\ c_0 + c_1\sigma - c_2 - c_3\sigma \\ c_0 - c_1 + c_2 - c_3 \\ c_0 - c_1\sigma - c_2 + c_3\sigma \end{pmatrix} \\ = \begin{pmatrix} C(1) - c_4 \\ C(\sigma) - c_4 \\ C(-1) - c_4 \\ C(-\sigma) - c_4 \end{pmatrix} &= \begin{pmatrix} A(1)B(1) - c_4 \\ A(\sigma)B(\sigma) - c_4 \\ A(-1)B(-1) - c_4 \\ A(-\sigma)B(-\sigma) - c_4 \end{pmatrix}, \end{aligned} \tag{2}$$

where $c_4 = a_2 b_2$. Let

$$P_0 = A(1)B(1), P_1 = A(\sigma)B(\sigma), P_2 = A(-1)B(-1), P_3 = A(-\sigma)B(-\sigma), \\ P_4 = c_4. \tag{3}$$

Then we arrive at

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = V_\sigma^{-1} \begin{pmatrix} P_0 - P_4 \\ P_1 - P_4 \\ P_2 - P_4 \\ P_3 - P_4 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -\sigma & -1 & \sigma \\ 1 & -1 & 1 & -1 \\ 1 & \sigma & -1 & -\sigma \end{pmatrix} \begin{pmatrix} P_0 - P_4 \\ P_1 - P_4 \\ P_2 - P_4 \\ P_3 - P_4 \end{pmatrix}$$

$$= \begin{pmatrix} P_0 + P_1 + P_2 + P_3 - P_4 \\ P_0 - P_1\sigma - P_2 + P_3\sigma \\ P_0 - P_1 + P_2 - P_3 \\ P_0 + P_1\sigma - P_2 - P_3\sigma \end{pmatrix}$$

using (2) and (3). In the above algorithm, the cost of the multiplication of V_σ and $(c_0, c_1, c_2, c_3)^T$ is virtually free. Thus, the above algorithm can compute a multiplication in $\mathbb{F}_{3^{6m}}$ by 5 multiplications in $\mathbb{F}_{3^{2m}}$, namely 15 multiplications in $\mathbb{F}_{3^m}$. Therefore, the algorithm of Gorla et al. [11] reduces the number of multiplications in $\mathbb{F}_{3^m}$. Note that the algorithm of Gorla et al. theoretically reaches the lower bound.

4 Proposed Residue Group Multiplication and Timing of the η_T Pairing

In this section, we present a residue group multiplication (RGM) algorithm in $\mathcal{G} = \mathbb{F}_{3^{6m}}^* / \mathbb{F}_{3^m}^*$. Its cost becomes 12 multiplications in $\mathbb{F}_{3^m}$ as $m \to \infty$, which reaches beyond the lower bound of the algorithm of Gorla et al. The proposed algorithm is effective when multiplication in the finite field is implemented using a basic method such as shift-and-add. Note that we can use RGMs at step 6 of Algorithm 1 due to what we described in Remark 1. Note that m is a prime for security of the pairing-based cryptography.

Moreover, we compared the timing of the η_T pairings using the algorithm of Gorla et al. and the proposed algorithm to verify whether the proposed algorithm is effective.

4.1 z Base

In the proposed RGM algorithm, $\mathbb{F}_{3^{6m}}$ directly represents the sixth extension of $\mathbb{F}_{3^m}$ unlike previous representations in Kerins et al. [13], Gorla et al. [11], and Beuchat et al. [3]. In other words, elements in $\mathbb{F}_{3^{6m}}$ are represented as polynomials with coefficients in $\mathbb{F}_{3^m}$ of one variable z. Although the m has to be co-prime to 6, it is satisfied because the m is a prime number. Consequently we can use a Vandermonde matrix (8×8) bigger than that of the algorithm of

Gorla et al. (4×4) (see (1)) to compute multiplications. The bigger Vandermond matrix reduces the cost of RGMs. Therefore, $\mathbb{F}_{3^{6m}}$ is represented as $\mathbb{F}_{3^m}[z]/k(z)$, where $k(z)$ is an irreducible polynomial with $k(z) = z^6 + z - 1$. Let V be an element in $\mathbb{F}_{3^{6m}}$. Then V can be represented by z base as follows:

$$V = v_5 z^5 + v_4 z^4 + v_3 z^3 + v_2 z^2 + v_1 z + v_0.$$

Then a set $\{z^5, z^4, z^3, z^2, z, 1\}$ forms a base of $\mathbb{F}_{3^{6m}}$ over $\mathbb{F}_{3^m}$. We call it z base in this paper. Let W be an element in $\mathbb{F}_{3^{6m}}$ represented by $\sigma\rho$ base with $W = w_5\sigma\rho^2 + w_4\rho^2 + w_3\sigma\rho + w_2\rho + w_1\sigma + w_0$. If $V = W$ then we can convert between V and W as follows:

$$\begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0 \\ 0&1&2&1&0&1 \\ 0&1&1&2&0&1 \\ 2&0&2&0&0&2 \\ 1&2&0&1&0&1 \\ 2&0&2&1&1&0 \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix}, \quad \begin{pmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&0&0 \\ 0&2&1&2&2&0 \\ 2&0&2&1&2&0 \\ 2&2&0&1&2&0 \\ 1&1&2&0&0&1 \\ 0&0&1&1&1&0 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{pmatrix}$$

$\sigma\rho$ base $\rightarrow$ z base $\qquad\qquad$ z base $\rightarrow$ $\sigma\rho$ base

Note that these conversions need no multiplications.

4.2 Proposed RGM Algorithm

The proposed RGM algorithm in $\mathcal{G} = \mathbb{F}^*_{3^{6m}} / \mathbb{F}^*_{3^m}$ uses the Vandermonde matrix in the same way as the algorithm of Gorla et al. [11]. Let $A(z), B(z)$ be elements in $\mathbb{F}_{3^{6m}}$ with $A(z) = a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0$ and $B(z) = b_5 z^5 + b_4 z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0$. Let $D(z)$ be an element in $\mathbb{F}_{3^m}[z]$ with

$$D(z) = A(z) \cdot B(z) = d_{10} z^{10} + d_9 z^9 + \cdots + d_1 z + d_0.$$

Note that there are relationships $d_0 = a_0 b_0$, $d_9 = a_4 b_5 + a_5 b_4$, and $d_{10} = a_5 b_5$. We then have to compute $D' = (d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8)$. Let $Z = (z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8) = (1, 2, x, x+1, x+2, -x, -(x+1), -(x+2))$, where x is the generator of the polynomial base of $\mathbb{F}_{3^m}$ over $\mathbb{F}_3$. Let V_z be the Vandermonde matrix for Z. Then we have the following matrix equation.

$$\begin{aligned} V_z D'^T &= \begin{pmatrix} z_1 & z_1^2 & \cdots & z_1^7 & z_1^8 \\ z_2 & z_2^2 & \cdots & z_2^7 & z_2^8 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_7 & z_7^2 & \cdots & z_7^7 & z_7^8 \\ z_8 & z_8^2 & \cdots & z_8^7 & z_8^8 \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_7 \\ d_8 \end{pmatrix} = \begin{pmatrix} D(z_1) - d_0 - d_9 z_1^9 - d_{10} z_1^{10} \\ D(z_2) - d_0 - d_9 z_2^9 - d_{10} z_2^{10} \\ \vdots \\ D(z_7) - d_0 - d_9 z_7^9 - d_{10} z_7^{10} \\ D(z_8) - d_0 - d_9 z_8^9 - d_{10} z_8^{10} \end{pmatrix} \\ &= \begin{pmatrix} A(z_1)B(z_1) - d_0 - d_9 z_1^9 - d_{10} z_1^{10} \\ A(z_2)B(z_2) - d_0 - d_9 z_2^9 - d_{10} z_2^{10} \\ \vdots \\ A(z_7)B(z_7) - d_0 - d_9 z_7^9 - d_{10} z_7^{10} \\ A(z_8)B(z_8) - d_0 - d_9 z_8^9 - d_{10} z_8^{10} \end{pmatrix} \end{aligned}$$

$$= \begin{pmatrix} A(1)B(1) - d_0 - d_9 - d_{10} \\ A(2)B(2) - d_0 + d_9 - d_{10} \\ \vdots \\ A(-(x+1))B(-(x+1)) - d_0 + d_9(x+1)^9 - d_{10}(x+1)^{10} \\ A(-(x+2))B(-(x+2)) - d_0 + d_9(x+2)^9 - d_{10}(x+2)^{10} \end{pmatrix}. \quad (4)$$

Let

$$\left.\begin{array}{ll} P_0 = d_0 = a_0 b_0, & P_1 = A(1)B(1), \\ P_2 = A(2)B(2), & P_3 = A(x)B(x), \\ P_4 = A(x+1)B(x+1), & P_5 = A(x+2)B(x+2), \\ P_6 = A(-x)B(-x), & P_7 = A(-(x+1))B(-(x+1)), \\ P_8 = A(-(x+2))B(-(x+2)), & P_9 = d_9 = a_4 b_5 + a_5 b_4, \\ P_{10} = d_{10} = a_5 b_5. & \end{array}\right\} \quad (5)$$

Using (4) and (5) we get

$$D'^T = V_z^{-1} \begin{pmatrix} P_1 - P_0 - P_9 - P_{10} \\ P_2 - P_0 + P_9 - P_{10} \\ P_3 - P_0 - P_9 x^9 - P_{10} x^{10} \\ P_4 - P_0 - P_9(x+1)^9 - P_{10}(x+1)^{10} \\ P_5 - P_0 - P_9(x+2)^9 - P_{10}(x+2)^{10} \\ P_6 - P_0 + P_9 x^9 - P_{10} x^{10} \\ P_7 - P_0 + P_9(x+1)^9 - P_{10}(x+1)^{10} \\ P_8 - P_0 + P_9(x+2)^9 - P_{10}(x+2)^{10} \end{pmatrix} = V_z^{-1} P'. \quad (6)$$

The matrix V_z^{-1} is explicitly represented as $\frac{1}{\beta}(\gamma_{ij})$, where $\beta = x^6 + x^4 + x^2 \in \mathbb{F}_{3^m}$ and each γ_{ij} is presented in Appendix A.

Recall that we may compute $(\beta d_0, \beta d_1, \cdots, \beta d_{10})$ instead of $(d_0, d_1, \cdots, d_{10})$ to compute $D(z) = A(z) \cdot B(z)$ in $\mathcal{G}$ due to Remark 1. Let $E(z) = \beta D(z) = e_{10} z^{10} + e_9 z^9 + \cdots + e_1 z + e_0$. We can compute coefficients of $E(z)$ as follows:

$$e_0 = \beta P_0,\ (e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8)^T = (\gamma_{ij})\, P',\ e_9 = \beta P_9,\ e_{10} = \beta P_{10}.$$

Next, we consider the cost of the proposed RGM algorithm in a number of multiplications. Let M be the cost of a multiplication in $\mathbb{F}_{3^m}$. Then we assume that the cost of multiplication in a polynomial of degree $(m-1)$ and a polynomial of degree k is $(\frac{k+1}{m})M$ with $m > k$ using the shift-and-add method. The cost of (5) is then $12M$. We will explain the cost of (6). First, let α_1 be the cost of computation of the vector P'. Multiplications that needs to compute P' are $P_9 x^9, P_9(x+1)^9, P_9(x+2)^9, P_{10} x^{10}, P_{10}(x+1)^{10}, P_{10}(x+2)^{10}$. The degree of P_9 and P_{10} is $(m-1)$. Then α_1 is $(\frac{10+10+10+11+11+11}{m})M = (\frac{63}{m})M$. Next, let α_2 be the cost of computation of $(\gamma_{ij})\, P'$. α_2 depends on the highest order of the column in (γ_{ij}). Then, α_2 is $(\frac{7+7+6+6+6+6}{m})M = (\frac{48}{m})M$. Last, let α_3 be costs

Table 1. Multiplication cost estimation

Multiplication method	The number of multiplications in $\mathbb{F}_{3^m}$
Gorla et al. algorithm [11]	15
Proposed RGM algorithm	$\mathbf{12 + \frac{132}{m}}$

Table 2. Timing on a Core 2 Duo E6320 1.86GHz ($m = 97$)

Multiplication method	Multiplication in $\mathbb{F}_{3^{6m}}$	η_T pairing
Gorla et al. algorithm [11]	73.2μs	5.01ms
Proposed RGM algorithm	**68.3μs**	**4.76ms**

of e_0, e_9, e_{10}. β is the polynomial of degree 6. Then α_3 is $(\frac{7+7+7}{m})M = (\frac{21}{m})M$. Therefore, the cost of a multiplication in the proposed algorithm is $(12 + \alpha_1 + \alpha_2 + \alpha_3)M = (12 + \frac{132}{m})M$. If $m \to \infty$, then the multiplication cost is $12M$.

The proposed algorithm is especially effective when multiplication in the finite field is implemented using a basic method, such as shift-and-add, so we used the shift-and-add method to estimate the cost of (6). We estimate multiplication costs in Table 1.

4.3 Timing of the η_T Pairing

Algorithm 2 is an algorithm for computing the η_T pairing modyfing Algorithm 1 with the propsed RGM alorithm and the z-base. We implemented the η_T pairing over $\mathbb{F}_{3^{97}}$ on a Core 2 Duo E6320 1.86GHz with 1GB RAM using gcc 3.4.4. We show the timing of the multiplication in $\mathbb{F}_{3^{6m}}$ and the η_T pairing in Table 2.

The number of multiplications for the proposed algorithm is $(12+\frac{132}{m})$. If $m = 97$, then the number of multiplications is $(12+\frac{132}{97}) = (13+\frac{35}{97})$. However, in the proposed RGM the number of additions increases by 212 for one multiplication compared to the algorithm of Gorla et al. in our implementation. Then the timing of the multiplication in $\mathbb{F}_{3^{6m}}$ is almost 7 percent faster than that of the multiplication algorithm by Gorla et al., which is 68.3μs. Moreover, the timing of the η_T pairing was almost 5 percent faster than that of the multiplication algorithm of Gorla et al., which is 4.76ms.

Remark 2.
The loop unrolling technique has been adopted to implement parings as [3], which can reduce the number of multiplications needed to compute a pairing. We can use the proposed algorithm together with the loop unrolling technique.

Algorithm 2 The η_T pairing algorithm using RGMs and the z base

INPUT: $P(x_p, y_p), Q(x_q, y_q) \in E(\mathbb{F}_{3^m})[r]$
OUTPUT: $\eta_T(P, Q) \in \mathbb{F}_{3^{6m}}$
1: $y_p \leftarrow -y_p, d \leftarrow 1$
2: $v \leftarrow x_p + x_q + 1$
3: $R_0 \leftarrow -y_p v + (y_q - y_p)z + (y_q + y_p)z^2 + y_p(x_p + x_q)z^3 - (y_p v + y_q)z^4 + y_p(x_p + x_q)z^5$
4: **for** $i \leftarrow 0$ to $(n-1)/2$ **do**
5: $v \leftarrow x_p + x_q + d$
6: $R_1 \leftarrow -v^2 + (y_p y_q + v)z + (y_p y_q - v)z^2 + v(v+1)z^3 - (v^2 + y_p y_q)z^4 + (v^2 + v + 1)z^5$
7: $R_0 \leftarrow R_0 R_1$ (**RGM**)
8: $y_p \leftarrow -y_p$
9: $x_q \leftarrow x_q^9, y_q \leftarrow y_q^9$
10: $d \leftarrow ((d-1) \bmod 3)$
11: $R_0 \leftarrow R_0^3$
12: **end for**
13: **return** R_0

5 Conclusion

In this study, we developed a residue group multiplication (RGM) algorithm in $\mathbb{F}^*_{3^{6m}} / \mathbb{F}^*_{3^m}$ to compute the η_T pairing. The proposed RGM algorithm takes $12 + \frac{132}{m}$ multiplications in $\mathbb{F}_{3^m}$, which reaches beyond the lower bound of the algorithm for multiplication in $\mathbb{F}^*_{3^{6m}}$ by Gorla et al. We can use the proposed RGM algorithm to compute the η_T pairing. Moreover, we implemented the η_T pairing on a Core 2 Duo E6320 1.86GHz with 1GB RAM using gcc 3.4.4 using the proposed RGM algorithm. The timing of the η_T pairing was almost 5 percent faster than that of the multiplication algorithm by Gorla et al., which is 4.76ms.

We expect that RGMs are applicable to other pairings, for example the Ate pairing using the Barreto-Naehrig curve [2], which has the embedding degree $k = 12$ defined over a large prime field.

References

1. P. Barreto, S. Galbraith, C. O'Eigeartaigh, and M. Scott, "Efficient pairing computation on supersingular Abelian varieties", Designs, Codes and Cryptography, Vol. 42, No. 3, pp. 239-271, 2007.
2. P. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime," *SAC 2005*, LNCS, Vol. 3897, pp. 319-331, 2006.
3. J.-L. Beuchat, N. Brisebarre, J. Detrey, E. Okamoto, M. Shirase, and T. Takagi, "Algorithms and arithmetic operators for computing the η_T pairing in characteristic three," IEEE Transactions on Computers, Vol. 57, No. 11, pp. 1454-1468, 2008.
4. M. Bodrato, "Towards optimal Toom-Cook multiplication for univariate and multivariate polynomials in characteristic 2 and 0," *WAIFI 2007*. LNCS, Vol. 4547, pp. 116-133, 2007.
5. D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," *EUROCRYPT 2004*, LNCS, Vol. 3027, pp. 506-522, 2004.

6. D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," *SIAM Journal of Computing*, Vol. 32, No. 3, pp. 586-615, 2003.
7. D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," *CRYPTO 2005*, LNCS, Vol. 3621, pp. 258-275, 2005.
8. R. Brent, P. Gaudry, E. Thomé, and P. Zimmermann, "Faster multiplication in GF(2)[x]," *ANTS 2008*, LNCS, Vol. 5011, pp. 153-166, 2008.
9. D. Cantor, "On arithmetical algorithms over finite fields," *J. Combinatorial Theory*, Series A-50, pp. 285-300, 1989.
10. S. Cook, "On the minimum computation time of functions," *PhD thesis*, Harvard University, 1966.
11. E. Gorla, C. Puttmann, and J. Shokrollahi, "Explicit formulas for efficient multiplication in $\mathbb{F}_{3^{6m}}$," *SAC 2007*, LNCS, Vol. 4876, pp. 173-183, 2007.
12. A. Karatsuba and Y. Ofman, "Multiplication of multidigit numbers on automata," Soviet Physics-Doklady, Vol. 7, pp. 595-596, 1963.
13. T. Kerins, W. P. Marnane, E. M. Popovici, and P. Barreto, "Efficient hardware for the Tate pairing calculation in characteristic three," *CHES 2005*, LNCS, Vol. 3659, pp. 412-426, 2005.
14. A. Lempel and S. Winograd, "A new approach to error-correcting codes," *IEEE Transactions on Information Theory* Vol. IT-23, pp. 503-508, 1977.
15. A. Schönhage, "Schnelle multiplikation von polynomial über K örpern der Charakteristik 2," *Acta Inf.* Vol. 7, pp. 395-398, 1977.
16. M. Shirase, T. Takagi, D. Choi, D.-H. Han, and H. Kim, "Efficient computation of Eta pairing over binary field with Vandermonde matrix," *ETRI Journal*, Vol. 31, No. 2, pp. 129-139, 2009.
17. A. Toom, "The complexity of a scheme of functional elements realizing the multiplication of integers," *Soviet Mathematics 3*, pp. 714-716, 1963.
18. S. Winograd, *Arithmetic complexity of computations*, SIAM, 1980.

A Elements of Matrix (γ_{ij})

When a matrix V_z^{-1} in (6) is represented as $V_z^{-1} = \frac{1}{\beta}(\gamma_{ij})$, each element γ_{ij} is provided as follows, where x is the generator of the base of $\mathbb{F}_{3^m}$ over $\mathbb{F}_3$ and $\beta = x^6 + x^4 + x^2$.

$$V_z^{-1} = \frac{1}{\beta}\begin{pmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & \gamma_{14} & \gamma_{15} & \gamma_{16} & \gamma_{17} & \gamma_{18} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & \gamma_{24} & \gamma_{25} & \gamma_{26} & \gamma_{27} & \gamma_{28} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & \gamma_{34} & \gamma_{35} & \gamma_{36} & \gamma_{37} & \gamma_{38} \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & \gamma_{44} & \gamma_{45} & \gamma_{46} & \gamma_{47} & \gamma_{48} \\ \gamma_{51} & \gamma_{52} & \gamma_{53} & \gamma_{54} & \gamma_{55} & \gamma_{56} & \gamma_{57} & \gamma_{58} \\ \gamma_{61} & \gamma_{62} & \gamma_{63} & \gamma_{64} & \gamma_{65} & \gamma_{66} & \gamma_{67} & \gamma_{68} \\ \gamma_{71} & \gamma_{72} & \gamma_{73} & \gamma_{74} & \gamma_{75} & \gamma_{76} & \gamma_{77} & \gamma_{78} \\ \gamma_{81} & \gamma_{82} & \gamma_{83} & \gamma_{84} & \gamma_{85} & \gamma_{86} & \gamma_{87} & \gamma_{88} \end{pmatrix}$$

$$\gamma_{11} = -(x^6 + x^4 + x^2) \qquad \gamma_{12} = x^6 + x^4 + x^2$$
$$\gamma_{21} = -(x^6 + x^4 + x^2) \qquad \gamma_{22} = -(x^6 + x^4 + x^2)$$
$$\gamma_{31} = 1 \qquad \gamma_{32} = -1$$
$$\gamma_{41} = 1 \qquad \gamma_{42} = 1$$
$$\gamma_{51} = 1 \qquad \gamma_{52} = -1$$
$$\gamma_{61} = 1 \qquad \gamma_{62} = 1$$
$$\gamma_{71} = 1 \qquad \gamma_{72} = -1$$
$$\gamma_{81} = 1 \qquad \gamma_{82} = 1$$

$$\gamma_{13} = -(x^5 + x^3 + x) \qquad \gamma_{14} = -(x^5 + 2x^4 + 2x^3 + x^2)$$
$$\gamma_{23} = -(x^4 + x^2 + 1) \qquad \gamma_{24} = -(x^4 + x^3 + x^2)$$
$$\gamma_{33} = x^5 \qquad \gamma_{34} = (x+1)^5$$
$$\gamma_{43} = x^4 \qquad \gamma_{44} = (x+1)^4$$
$$\gamma_{53} = x^3 \qquad \gamma_{54} = (x+1)^3$$
$$\gamma_{63} = x^2 \qquad \gamma_{64} = (x+1)^2$$
$$\gamma_{73} = x \qquad \gamma_{74} = (x+1)$$
$$\gamma_{83} = 1 \qquad \gamma_{84} = 1$$

$$\gamma_{15} = -(x^5 + x^4 + 2x^3 + 2x^2) \qquad \gamma_{16} = x^5 + x^3 + x$$
$$\gamma_{25} = -(x^4 + 2x^3 + x^2) \qquad \gamma_{26} = -(x^4 + x^2 + 1)$$
$$\gamma_{35} = (x+2)^5 \qquad \gamma_{36} = -x^5$$
$$\gamma_{45} = (x+2)^4 \qquad \gamma_{46} = x^4$$
$$\gamma_{55} = (x+2)^3 \qquad \gamma_{56} = -x^3$$
$$\gamma_{65} = (x+2)^2 \qquad \gamma_{66} = x^2$$
$$\gamma_{75} = (x+2) \qquad \gamma_{76} = -x$$
$$\gamma_{85} = \qquad \gamma_{86} = 1$$

$$\gamma_{17} = x^5 + 2x^4 + 2x^3 + x^2 \qquad \gamma_{18} = x^5 + x^4 + 2x^3 + 2x^2$$
$$\gamma_{27} = -(x^4 + x^3 + x^2) \qquad \gamma_{28} = -(x^4 + 2x^3 + x^2)$$
$$\gamma_{37} = -(x+1)^5 \qquad \gamma_{38} = -(x+2)^5$$
$$\gamma_{47} = (x+1)^4 \qquad \gamma_{48} = (x+2)^4$$
$$\gamma_{57} = -(x+1)^3 \qquad \gamma_{58} = -(x+2)^3$$
$$\gamma_{67} = (x+1)^2 \qquad \gamma_{68} = (x+2)^2$$
$$\gamma_{77} = -(x+1) \qquad \gamma_{78} = -(x+2)$$
$$\gamma_{87} = 1 \qquad \gamma_{88} = 1$$