Universal η_T Pairing Algorithm over Arbitrary Extension Degree

Masaaki Shirase[1], Yuto Kawahara[1], Tsuyoshi Takagi[1], and Eiji Okamoto[2]

[1] Future University-Hakodate, Japan
[2] University of Tsukuba, Japan

Abstract. The η_T pairing on supersingular is one of the most efficient algorithms for computing the bilinear pairing [3]. The η_T pairing defined over finite field $\mathbb{F}_{3^n}$ has embedding degree 6, so that it is particularly efficient for higher security with large extension degree n. Note that the explicit algorithm over $\mathbb{F}_{3^n}$ in [3] is designed just for $n \equiv 1 \pmod{12}$, and it is relatively complicated to construct an explicit algorithm for $n \not\equiv 1 \pmod{12}$. It is better that we can select many n's to implement the η_T pairing, since n corresponds to security level of the η_T pairing.
In this paper we construct an explicit algorithm for computing the η_T pairing with arbitrary extension degree n. However, the algorithm should contain many branch conditions depending on n and the curve parameters, that is undesirable for implementers of the η_T pairing. This paper then proposes the universal η_T pairing ($\widetilde{\eta_T}$ pairing), which satisfies the bilinearity of pairing (compatible with Tate pairing) without any branches in the program, and is as efficient as the original one. Therefore the proposed universal η_T pairing is suitable for the implementation of various extension degrees n with higher security.

Keywords: Tate pairing, η_T pairing, Duursma-Lee algorithm, efficient implementation.

1 Introduction

Recently, bilinear pairings defined on elliptic curves such as Tate pairing and the η_T pairing have been attracted to make new cryptographic protocols, for example, identity-based cryptosystem [5], short signature [7] and efficient broadcast cryptosystem [6], come true.

A standard algorithm for computing the Tate pairing is Miller algorithm [12]. The computational cost of Miller algorithm is generally larger than that of RSA or elliptic curve cryptosystems [2]. It is one of important research fields in cryptography to improve the computational cost of pairings. Supersingular curves with characteristic three has embedding degree 6, so that it is particularly efficient for higher security. Some efficient variations of Miller algorithm in base three have been proposed for computing Tate pairing on supersingular elliptic curves over characteristic three [2, 10]. Duursma and Lee proposed a closed form generated by divisor $g_R = 3(R)+(-3R)-4(\mathcal{O})$ for a point R, which can efficiently compute Tate pairing [8]. Barreto *et. al.* then proposed the η_T pairing which can

reduce the iteration number of the main loop of Duursma-Lee algorithm [3]. The computational cost of the η_T pairing is about half of the Duursma-Lee algorithm. The η_T pairing is currently one of the fastest algorithm for computing the bilinear pairing. It is easy to convert between Tate pairing and the η_T pairing (see [3] or [4] for details).

This paper focuses on the η_T pairing defined over finite field $\mathbb{F}_{3^n}$. Extension degree n of $\mathbb{F}_{3^n}$ has to satisfy the following conditions due to several attacks: n is an odd prime number, l is a large prime number with $l|(3^{6n}-1)$, where l is the order of the subgroup of the elliptic curve used in pairing. The extension degrees that satisfy these conditions are $n = 97, 163, 167, 193, 239, 313, 353, \ldots$. On the other hand the explicit algorithm for computing the η_T pairing in [3] deals only with $n \equiv 1 \pmod{12}$. Therefore, the previous researches on the η_T pairing have been implemented in the case of $n \equiv 1 \pmod{12}$ [3, 4, 14]. To our knowledge there is no literature that proposes the η_T pairing over $\mathbb{F}_{3^n}$ for general extension degree n.[1] Note that we should modify it if we try to construct an explicit algorithm for $n \not\equiv 1 \pmod{12}$, namely $n = 163, 167, 239, 353, \ldots$. It is relatively complicated to construct an explicit algorithm for $n \not\equiv 1 \pmod{12}$.

In this paper we present an explicit algorithm for arbitrary prime number n with $\gcd(n, 6) = 1$. The proposed explicit algorithm depends on the extension degree n and the coefficients of the underlying curves, which is not suitable for implementers of the η_T pairing. Therefore this paper proposes the universal η_T pairing whose algorithm does not depend on n and whose computational cost is same as the original η_T pairing. Moreover we present the explicit relationship between Tate pairing and the universal η_T pairing, which make the universal η_T pairing compatible with Tate pairing for arbitrary extension degree n.

The remainder of this paper is organized as follows: In Section 2 we explain about the known properties of the η_T pairing. In Section 3 we describe the proposed algorithms including an explicit algorithm for computing the η_T pairing over arbitrary degree n and the universal η_T pairing. Proposition 1 shows the relationship between Tate pairing and the universal η_T pairing. We then present some timings of the universal η_T pairing in C language. In Section 4 we present the proof of Proposition 1 and the correctness of algorithms appeared in Section 3. In Section 5 we conclude this paper.

2 Tate pairing over supersingular curve with characteristic three

Let $\mathbb{F}_{3^n}$ be an extension field over $\mathbb{F}_3$ of degree n. Let E^b be the supersingular elliptic curve defined by $y^2 = x^3 - x + b$ with $b \in \{1, -1\}$. All supersingular curves are isomorphic to this curve. The set of all points on E^b over $\mathbb{F}_{3^n}$ defined by

$$E^b(\mathbb{F}_{3^n}) = \{(x, y) \in \mathbb{F}_{3^n} \times \mathbb{F}_{3^n} \ : \ y^2 = x^3 - x + b\} \cup \{\mathcal{O}\},$$

[1] In the case of the η_T pairing over $\mathbb{F}_{2^n}$, MIRACL supports the general extension degree using 4 branches [13].

forms a group, where $\mathcal{O}$ is the point at infinity. Note that the extension degree n should be $gcd(n, 6) = 1$, it then satisfies $n \equiv 1, 5, 7, 11 \pmod{12}$ [3]. In this paper we deal with the arbitrary degree n with $gcd(n, 6) = 1$. We define b' as

$$b' = \begin{cases} b & \text{if } n \equiv 1,\ 11 \pmod{12}, \\ -b & \text{if } n \equiv 5,\ 7 \pmod{12}, \end{cases} \tag{1}$$

then it is known that

$$\#E^b(\mathbb{F}_{3^n}) = 3^n + 1 + b'3^{(n+1)/2}. \tag{2}$$

2.1 Tate Pairing

Let l be a large prime number, $l \mid \#E^b(\mathbb{F}_{3^n})$ and $l \mid (3^{6n} - 1)$. Let $P \in E^b(\mathbb{F}_{3^n})[l]$ and let $Q \in E^b(\mathbb{F}_{3^{6n}})/lE^b(\mathbb{F}_{3^{6n}})$. Then Tate pairing $e(P, Q)$ over $E^b(\mathbb{F}_{3^n})$ is a pairing, $e : E^b(\mathbb{F}_{3^n})[l] \times E^b(\mathbb{F}_{3^{6n}})/lE^b(\mathbb{F}_{3^{6n}}) \to \mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^l$, and defined as $e(P, Q) = f_{P,l}(Q)$, where $f_{P,N}$ is a function whose divisor is $(f_{P,N}) = (N - 1)(P) - ((N - 1)P) - (N - 2)(\mathcal{O})$ for any positive integer N.

Since $e(P, Q) \in \mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^l$, we require an arithmetic on $\mathbb{F}_{3^{6n}}$. A basis $\{1, \sigma, \rho, \sigma\rho, \rho^2, \sigma\rho^2\}$ of $\mathbb{F}_{3^{6n}}$ over $\mathbb{F}_{3^n}$ gives an efficient arithmetic on $\mathbb{F}_{3^{6n}}$, where σ and ρ satisfy $\sigma = -1$ and $\rho^3 = \rho + b$.

For a point $Q = (x, y) \in E^b(\mathbb{F}_{3^n})$ the distortion map ψ is one-to-one homomorphism defined by

$$\psi(x, y) = (\rho - x, y\sigma) \text{ in } E^b(\mathbb{F}_{3^{6n}}). \tag{3}$$

Then $e(P, \psi(Q))$ is defined for $P, Q \in E^b(\mathbb{F}_{3^n})$. Note that the representation of $e(P, \psi(Q))$ has ambiguity since $e(P, \psi(Q))$ is contained in a coset of the residue group $\mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^l$. In order to remove this ambiguity, the final exponentiation is required, which is a powering by $(3^{6n} - 1)/l$. Here we denote $e(P, \psi(Q))^{(3^{6n}-1)/l}$ by $\hat{e}(P, Q)$, then $\hat{e}(P, Q)$ has bilinearity, namely $\hat{e}(aP, Q) = \hat{e}(P, aQ) = \hat{e}(P, Q)^a$ for any non zero integer a. The bilinearity is used in many new cryptographic applications such as identity-based cryptosystem [5], short signature [7] and efficient broadcast cryptosystem [6].

Miller proposed an efficient algorithm for computing $f_{P,l}(\psi(Q))$ on arbitrary elliptic curve over arbitrary field [12]. Barreto *et. al.* [2] and Galbraith *et. al.* [10] proposed Miller algorithm in base three using the following calculation of function f at point $Q \in E^b(\mathbb{F}_{3^n})$, $f \leftarrow f^3 \cdot (l_1 l_2)(Q)$, where l_1, l_2 are a tangent line of E^b at Q and a line going through Q and $2Q$, respectively.

Miller algorithm in base three is suitable for pairing on $E^b(\mathbb{F}_{3^n})$ since cubing operation and a computation of $3Q$ are virtually for free. Note that $3Q$ for $Q = (x_q, y_q) \in E^b(\mathbb{F}_{3^n})$ is calculated as follows:

$$3Q = (x_q^9 - b, -y_q^9) = \phi\pi^2(Q), \tag{4}$$

where π is the 3rd-power Frobenius map on E^b, namely $\pi(Q) = (x_q^3, y_q^3)$, and ϕ is a map defined as

$$\phi(x_q, y_q) = (x_q - b, -y_q). \tag{5}$$

2.2 Duursma-Lee Algorithm

There is an important property of Tate pairing [10]. Let m be an integer such that $l \mid m$ and $m \mid (3^{6n}-1)$. Then $f_{P,m}(\psi(Q))^{(3^{6n}-1)/m} = f_{P,l}(\psi(Q))^{(3^{6n}-1)/l} = \hat{e}(P,Q)$.

Duursma and Lee effectively used this property to propose a closed algorithm for computing Tate pairing on supersingular curves [8]. We know that $l \mid (3^{3n}+1)$ and $(3^{3n}+1) \mid (3^{6n}-1)$ due to Eq.(2) and $l|\#E(\mathbb{F}_{3^n})$. In the algorithm $3^{3n}+1$ is set to N, where the Hamming weight of N in base three is very sparse. Duursma and Lee then showed that the function $l_1 l_2$ of Miller algorithm in base three is equivalent to an explicit function

$$g_R(x,y) = y_r^3 y - (x_r^3 + x - b)^2, \tag{6}$$

whose divisor is $(g_R) = 3R + (-3R) - 4(\mathcal{O})$ for $R = (x_r, y_r)$.

The function g_R can be utilized to compute a function $f_{P,3^k+1}$ for any positive integer k [11],

$$f_{P,3^k+1} = g_P^{3^{k-1}} g_{3P}^{3^{k-2}} \cdots g_{3^{k-2}P}^{3} g_{3^{k-1}P}. \tag{7}$$

Setting $k = 3n$ in Eq.(7), we see $f_{P,3^{3n}+1}(\psi(Q)) = \prod_{i=1}^{3n} (g_{3^{i-1}P}(\psi(Q)))^{3^{3n-i}}$. Therefore we obtain

$$f_{P,3^{3n}+1}(\psi(Q)) = \prod_{i=1}^{n} g_{\pi^i(P)}(\pi^{n+1-i}(\psi(-Q)) \tag{8}$$

due to $(g_{3^{i-1}P}(\psi(Q)))^{3^{3n-i}} = (g_{3^{i-n-1}P}(\psi(Q)))^{3^{2n-i}} = (g_{3^{i-2n-1}P}(\psi(Q)))^{3^{n-i}}$. The explicit description of Duursma-Lee algorithm is derived from Eq. (8) and thus it has n iterations in the main loop.

2.3 η_T Pairing

Barreto *et. al.* [3] proposed the η_T pairing to decrease the iteration number of Duursma-Lee algorithm. Here we describe the η_T pairing on supersingular curve over characteristic three. Let T be an integer such that

$$T = 3^{(n+1)/2} + b'. \tag{9}$$

Then the $\eta_T(P,Q)$ for $P, Q \in E^b(\mathbb{F}_{3^n})$ is defined as $\eta_T(P,Q) = f_{-P,T}(\psi(Q))$ if $b' = 1$ and $\eta_T(P,Q) = f_{P,T}(\psi(Q))$ otherwise.

Setting $k = (n+1)/2$ in Eq.(7), we see $f_{P,3^{(n+1)/2}+1} = \prod_{i=1}^{(n+1)/2} g_{3^{i-1}}^{3^{(n+1)/2-i}}$. Barreto *et. al.* showed that the difference between $f_{\pm P,T}$ and $f_{P,3^{(n+1)/2}+1}$ is represented by a function of a line $l_{3P',b'P}$ going through $3P'$ and $b'P$, where $P' = 3^{(n-1)/2}P$. Then $\eta_T(P,Q) = l_{3P',b'P}(\psi(Q)) \prod_{j=0}^{(n-1)/2} g_{3^iP}(\psi(Q))^{3^{(n-1)/2-i}}$. Moreover it can be rewritten as

$$\eta_T(P,Q) = l_{3P',b'P}(\psi(Q)) \prod_{j=0}^{(n-1)/2} g_{3^{-j}P'}(\psi(Q))^{3^j}, \tag{10}$$

to remove the exponent $3^{(n-1)/2}$.

Eq.(10) is similar to Eq.(8), but only has $(n+1)/2$ iterations, which means the cost of η_T pairing is about half of Duursma-Lee algorithm. Note that $T = 3^{(n+1)/2} \pm 1$ is as large as $|\#E^b(\mathbb{F}_{3^n}) - 3^n - 1|$, which is the absolute value of the trace of $E^b(\mathbb{F}_{3^n})$.

$\eta_T(P,Q)$ itself is contained in a coset of the residue group $\mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^{\#E^b(\mathbb{F}_{3^n})}$. Therefore one cannot use $\eta_T(P,Q)$ in cryptographic protocols due to its ambiguity. $\eta_T(P,Q)$ requires the final exponentiation of powering by W to be a bilinear pairing, where W is an integer defined as

$$W = (3^{3n}-1)(3^n+1)(3^n+1-b'(3^{(n+1)}))\,(= (3^{6n}-1)/\#E^b(\mathbb{F}_{3^n})\,). \quad (11)$$

There is an efficient algorithm for computing the final exponentiation in [15].

Let Z be an integer such that

$$Z = -b'3^{(n+3)/2}. \quad (12)$$

Then there is a relationship between the η_T pairing and Tate pairing,

$$(\eta_T(P,Q)^W)^{3T^2} = \hat{e}(P,Q)^Z. \quad (13)$$

It is essential to find an algorithm for computing $\hat{e}(P,Q)^X$ for some integer X that becomes a bilinear pairing. However, if we need to convert the η_T pairing to Tate pairing via Eq.(13), there is an efficient conversion algorithm, see [4].

Note that the original algorithm for computing the η_T pairing in [3] includes computations of cube root computations. In general it takes the cost of 0.8∼2 multiplications [1], then we cannot neglect their costs. Beuchat *et. al.* generated an algorithm (Algorithm 2 in [4]) that has no cube root and outputs $\eta_T(P,Q)^{3^{(n+1)/2}}$.

3 Proposed Explicit Algorithms

In this section we present an explicit algorithm for computing the η_T pairing with arbitrary extension degree n. We then propose the universal η_T pairing whose algorithm has no branch in the program.

3.1 η_T Pairing for Arbitrary n

An algorithm for computing the η_T pairing with arbitrary extension degree n can be constructed from Eq.(10). Since both $l_{3P',b'P}$ and $g_{3^{-j}P'}$ depend on the extension degree n and the curve parameter b', the explicit description of η_T pairing has a complex form and causes many branches in the program. Lemma 5 of [3] explains about $l_{3P',b'P}$ in all cases, however $g_{3^{-j}P'}$ is considered only for $n \equiv 1 \pmod{12}$. In this section we investigate $g_{3^{-j}P'}$ in details.

Note that $g_{3^{-j}P'}$ needs a computation of $P' = 3^{(n-1)/2}P$. In order to efficiently compute P' we use Eq.(4), then we see

$$P' = \phi^{(n-1)/2}\pi^{(n-1)}(P). \quad (14)$$

Algorithm 1 : Computation of $\eta_T(P,Q)^{3^{(n+1)/2}}$ for arbitrary n

input: $P=(x_p,y_p), Q=(x_q,y_q) \in E^b(\mathbb{F}_{3^n})$

output: $(\eta_T(P,Q))^{3^{(n+1)/2}} \in \mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^{\#E^b(\mathbb{F}_{3^n})}$

1. $b' \leftarrow \begin{cases} b & \text{if } n \equiv 1,\ 11 \pmod{12} \\ -b & \text{if } n \equiv 5,\ 7 \pmod{12} \end{cases}$
2. **if** $b'=1$ **then** $y_p \leftarrow -y_p$
3. $R_0 \leftarrow \begin{cases} -y_p(x_p+x_q+b)+y_q\sigma+y_p\rho & \text{if } n \equiv 1 \pmod{12} \\ -y_p(x_p+x_q-b)+y_q\sigma+y_p\rho & \text{if } n \equiv 5 \pmod{12} \\ y_p(x_p+x_q+b)+y_q\sigma-y_p\rho & \text{if } n \equiv 7 \pmod{12} \\ y_p(x_p+x_q-b)+y_q\sigma-y_p\rho & \text{if } n \equiv 11 \pmod{12} \end{cases}$
4. $d \leftarrow \begin{cases} b & \text{if } n \equiv 1,7 \pmod{12} \\ -b & \text{if } n \equiv 5,11 \pmod{12} \end{cases}$
5. **for** $i \leftarrow 0$ **to** $(n-1)/2$ **do**
6. $\quad r_0 \leftarrow x_p+x_q+d$
7. $\quad R_1 \leftarrow \begin{cases} -r_0^2+y_py_q\sigma-r_0\rho-\rho^2 & \text{if } n \equiv 1,5 \pmod{12} \\ -r_0^2-y_py_q\sigma-r_0\rho-\rho^2 & \text{if } n \equiv 7,11 \pmod{12} \end{cases}$
8. $\quad R_0 \leftarrow R_0R_1$
9. $\quad y_q \leftarrow -y_p$
10. $\quad x_q \leftarrow x_q^9,\ y_q \leftarrow y_q^9$
11. $\quad R_0 \leftarrow R_0^3$
12. $\quad d \leftarrow d-b \pmod 3$
13. **end for**
14. **return** R_0

The explicit description of P' depends on not only the extension degree n but also the curve parameter b arisen from ϕ in Eq. (5).

We present Algorithm 1 which is an explicit algorithm for computing the η_T pairing with arbitrary n. The proposed explicit algorithm is based on the variation of the η_T pairing discussed by Beuchat *et. al.* [4] which has no cube root computation for $n \equiv 1 \pmod{12}$. Refer Section 4.1 for a proof of the correctness of Algorithm 1.

The branches in Steps 1-4 and Step 7 are caused by $l_{3P',b'P}$ (Lemma 5 of [3]) and $g_{3-jP'}$, respectively.

3.2 Universal η_T Pairing

Algorithm 1 has many branches that depend on the value of (n mod 12) and b'. If there is an algorithm without branches, then it becomes more implementor-friendly. Therefore Section 3.2 proposes the universal η_T pairing, $\widetilde{\eta_T}(P,Q)$, that has no branch and is as efficient as the original η_T pairing. The proposed algorithm is given by Algorithm 2.

The following proposition describes the difference between the η_T pairing (Algorithm 1) and the $\widetilde{\eta_T}$ pairing (Algorithm 2).

Algorithm 2 : Computation of $\widetilde{\eta_T}(P,Q)$ for arbitrary n

input: $P=(x_p,y_p), Q=(x_q,y_q) \in E^b(\mathbb{F}_{3^n})$
output: $\widetilde{\eta_T}(P,Q) \in \mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^{\#E^b(\mathbb{F}_{3^n})}$

1. $R_0 \leftarrow -y_p(x_p+x_q+b)+y_q\sigma+y_p\rho$
2. $d \leftarrow b$
3. **for** $i \leftarrow 0$ **to** $(n-1)/2$ **do**
4. $\quad r_0 \leftarrow x_p+x_q+d$
5. $\quad R_1 \leftarrow -r_0^2+y_py_q\sigma-r_0\rho-\rho^2$
6. $\quad R_0 \leftarrow R_0R_1$
7. $\quad y_q \leftarrow -y_p$
8. $\quad x_q \leftarrow x_q^9,\ y_q \leftarrow y_q^9$
9. $\quad R_0 \leftarrow R_0^3$
10. $\quad d \leftarrow d-b \pmod 3$
11. **end for**
12. **return** R_0

Proposition 1. *Let n be an odd prime with $gcd(n,6)=1$, and let T, W and Z be integers defined as Eqs.(9), (11) and (12), respectively. Then we have the following properties of $\widetilde{\eta_T}(P,Q)$ for $P,Q \in E^b(\mathbb{F}_{3^n})$.*
(i) $\widetilde{\eta_T}(P,Q)^W$ with final exponentiation W is a non-degenerate and bilinear pairing.
(ii) $\widetilde{\eta_T}(P,Q)^W=\hat{e}(P,Q)^U$, where $U=(3^{(n-1)/2}\cdot VZT^{-2} \bmod \#E^b(\mathbb{F}_{3^n}))$ and V is defined by the following table.

	$b=1$	$b=-1$
$n \equiv 1$ *(mod 12)*	-1	1
$n \equiv 5$ *(mod 12)*	$3^{(n+1)/2}-2$	$3^{(n+1)/2}+2$
$n \equiv 7$ *(mod 12)*	-1	1
$n \equiv 11$ *(mod 12)*	$-3^{(n+1)/2}-2$	$-3^{(n+1)/2}+2$

The proof of Proposition 1 is described in Section 4.2. The final exponentiation for $\widetilde{\eta_T}(P,Q)$ is same as that for the $\eta_T(P,Q)$ pairing, which is efficiently computed by the algorithm from [15]. Due to Proposition 1-(i) we can apply the $\widetilde{\eta_T}$ pairing to cryptographic applications with a bilinear pairing. If necessary, we can obtain Tate pairing $\hat{e}(P,Q)$ from $\widetilde{\eta_T}(P,Q)$ due to Proposition 1-(ii).

Note that $\widetilde{\eta_T}(P,Q)^W$ is included in the torus $T_2(\mathbb{F}_{3^{3n}})$. Therefore the conversion of $\widetilde{\eta_T}(P,Q)^W$ to $\hat{e}(P,Q)$, a powering by U^{-1}, can be efficiently performed with arithmetic in $T_2(\mathbb{F}_{3^{3n}})$, refer to [15].

Moreover, the proposed $\widetilde{\eta_T}$ pairing has good properties, namely it has no branch and no cube root computation unlike the original η_T pairing. The $\widetilde{\eta_T}$ pairing is as efficient as the variation of η_T pairing, which is one of the fastest implementations of a bilinear pairing [4].

3.3 Implementation Results

We implemented the $\widetilde{\eta_T}$ pairing (Algorithm 2) in C language. It is implemented on an AMD Opteron$^{\mathrm{TM}}$ Processor 275 at 2.2GHz using 8GByte RAM.

We mainly follow the implementation described in [9]. The polynomial base representation is used for $\mathbb{F}_{3^n}$. Finite field $\mathbb{F}_3 = \{0, 1, 2\}$ is encoded by two bits, and an addition in $\mathbb{F}_3$ is programmed by 7 Boolean logic operations [9]. We implemented the multiplication by the right-to-left sfift-addition algorithm with the signed window method of width 3. The extended Euclidean algorithm is used for the inversion. We deploy the final exponentiation using the torus proposed by Shirase *et. al.* [15].

Table 1. Timing of operations on $\mathbb{F}_{3^n}$ and computation of the $\widetilde{\eta_T}$ pairing (μsec)

Extension degree (n)	97(SSE)	97	167	193	239	313
Addition	0.0083	0.0168	0.0210	0.0237	0.0265	0.0377
Cubing	0.0394	0.1610	0.2104	0.2694	0.3052	0.3943
Multiplication	0.5009	1.2056	2.9757	3.7164	5.3137	8.2219
Inversion	7.7111	12.0865	28.6980	39.7646	55.5295	95.9911
$\widetilde{\eta_T}^W$ (Alg.2+[15])	479.63	1164.16	4406.26	6267.99	10753.17	21796.96

Table 1 presents the timing of the $\widetilde{\eta_T}$ pairing for different extension degrees $n = 97, 167, 193, 239, 313$. The timing is an average value for 1,000,000 randomly chosen elements on the base field $\mathbb{F}_{3^n}$ or elliptic curve $E^b(\mathbb{F}_{3^n})$. If we choose about twice larger extension degree, then the $\widetilde{\eta_T}$ pairing becomes about 5 times slower. The $\widetilde{\eta_T}$ pairing with $n = 313$ can be implemented in about 20 milliseconds. In case of $n = 97$ we optimized our programming suitable for the streaming SIMD extensions (SSE). The timing using SSE for the $\widetilde{\eta_T}$ pairing with $n = 97$ achieves under 0.5 milliseconds, which is more than twice as fast than the implementation without SSE. The embedded field of extension degree n is $\mathbb{F}_{3^{6n}}$, and their bit size are 923, 1589, 1836, 2273, 2977 for $n = 97,\ 167,\ 193,\ 239,\ 313$, respectively.

4 Proofs of Proposition and Algorithm

We prove the Proposition 1 and the correctness of Algorithm 1 described in this paper.

4.1 Proof of Algorithm 1

In order to prove the correctness of Algorithm 1 we introduce Algorithm 3 which is an extension of the original $\eta_T(P, Q)$ [3] to arbitrary extension degree n. Denote by $R_{0,j}^{(Alg.1)}$ and $R_{0,j}^{(Alg.3)}$ the value in register R_0 at the j-th

Algorithm 3 : Computation of $\eta_T(P,Q)$ for arbitrary n (Including cube root version)

input: $P=(x_p,y_p), Q=(x_q,y_q) \in E^b(\mathbb{F}_{3^n})$

output: $\eta_T(P,Q) \in \mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^{\#E^b(\mathbb{F}_{3^n})}$

1. $b' \leftarrow \begin{cases} b & \text{if } n \equiv 1,\ 11 \pmod{12} \\ -b & \text{if } n \equiv 5,\ 7 \pmod{12} \end{cases}$
2. **if** $b'=1$ **then** $y_p \leftarrow -y_p$
3. $R_0 \leftarrow \begin{cases} -y_p(x_p+x_q+b)+y_q\sigma+y_p\rho & \text{if } n \equiv 1 \pmod{12} \\ -y_p(x_p+x_q-b)+y_q\sigma+y_p\rho & \text{if } n \equiv 5 \pmod{12} \\ y_p(x_p+x_q+b)+y_q\sigma-y_p\rho & \text{if } n \equiv 7 \pmod{12} \\ y_p(x_p+x_q-b)+y_q\sigma-y_p\rho & \text{if } n \equiv 11 \pmod{12} \end{cases}$
4. **for** $i \leftarrow 0$ **to** $(n-1)/2$ **do**
5. $\quad r_0 \leftarrow \begin{cases} x_p+x_q+b \text{ if } n \equiv 1,7 \pmod{12} \\ x_p+x_q-b \text{ if } n \equiv 5,11 \pmod{12} \end{cases}$
6. $\quad R_1 \leftarrow \begin{cases} -r_0^2+y_py_q\sigma-r_0\rho-\rho^2 & \text{if } n \equiv 1,5 \pmod{12} \\ -r_0^2-y_py_q\sigma-r_0\rho-\rho^2 & \text{if } n \equiv 7,11 \pmod{12} \end{cases}$
7. $\quad R_0 \leftarrow R_0R_1$
8. $\quad x_p \leftarrow x_p^{1/3},\ y_p \leftarrow y_p^{1/3}$
9. $\quad x_q \leftarrow x_q^3,\ y_q \leftarrow y_q^3$
10. **end for**
11. **return** R_0

loop of Algorithm 1 and Algorithm 3, respectively. They are related by equation $R_{0,j}^{(Alg.1)} = (R_{0,j}^{(Alg.3)})^{3^{j+1}}$ (see also Appendix II in [4]). Therefore we see that if Algorithm 3 outputs $\eta_T(P,Q)$ then Algorithm 1 outputs $\eta_T(P,Q)^{3^{(n+1)/2}}$. Then it is sufficient to prove the correctness of Algorithm 3.

Recall that $\eta_T(P,Q)$ for arbitrary n is defined using two values, $l_{3P',b'P}(\psi(Q))$ and $g_{3^{-j}P'}(\phi(Q))^{3^j}$. We prove that $g_{3^{-j}P'}(\phi(Q))^{3^j}$, which corresponds to the j-th loop of Step 5 and 6 in Algorithm 3, can be computed by Lemma 1.

Lemma 1. *Let n be an odd prime. Then*

$$g_{3^{-j}P'}(\phi(Q))^{3^j} = \begin{cases} -r_0^2+y_p^{(-j)}y_q^{(j)}\sigma-r_0\rho-\rho^2 & \textit{if } n \equiv 1 \textit{ (mod 4)}, \\ -r_0^2-y_p^{(-j)}y_q^{(j)}\sigma-r_0\rho-\rho^2 & \textit{if } n \equiv 3 \textit{ (mod 4)}, \end{cases}$$

for $P=(x_p,y_p),\ Q=(x_q,y_q) \in E^b(\mathbb{F}_{3^n})$, where r_0 is defined as

$$r_0 = \begin{cases} x_p+x_q+b \textit{ if } n \equiv 1 \textit{ (mod 6)}, \\ x_p+x_q-b \textit{ if } n \equiv 5 \textit{ (mod 6)}. \end{cases}$$

Proof. See the appendix. □

Next we have

$$l_{3P',b'P}(x,y) = \begin{cases} y+y_p(x-x_p)-b'y_p & \text{if } n \equiv 1,5 \pmod{12}, \\ y-y_p(x-x_p)-b'y_p & \text{if } n \equiv 7,11 \pmod{12}. \end{cases} \tag{15}$$

Algorithm 4 : Computation of $\overline{\eta_T}(P,Q)$ for arbitrary n

input: $P=(x_p,y_p), Q=(x_q,y_q) \in E^b(\mathbb{F}_{3^n})$
output: $\overline{\eta_T}(P,Q) \in \mathbb{F}^*_{3^{6n}}/(\mathbb{F}^*_{3^{6n}})^{\#E^b(\mathbb{F}_{3^n})}$
1. $R_0 \leftarrow -y_p(x_p+x_q+b)+y_q\sigma+y_p\rho$
2. **for** $i \leftarrow 0$ **to** $(n-1)/2$ **do**
3. $\quad r_0 \leftarrow x_p+x_q+b$
4. $\quad R_1 \leftarrow -r_0^2+y_py_q\sigma-r_0\rho-\rho^2$
5. $\quad R_0 \leftarrow R_0R_1$
6. $\quad x_p \leftarrow x_p^{1/3},\ y_p \leftarrow y_p^{1/3}$
7. $\quad x_q \leftarrow x_q^3,\ y_q \leftarrow y_q^3$
8. **end for**
9. **return** R_0

from Lemma 5 of [3]. Therefore the formula for R_0 in Steps 1 and 3 can be obtained due to Eqs. (1) and (15). Therefore we prove the correctness of Algorithm 3.

4.2 Proof of Proposition 1

We first prove Proposition 1-(ii). Let $\overline{\eta_T}(P,Q)$ be the output of Algorithm 4. Denote by $R_{0,j}^{(Alg.2)}$ and $R_{0,j}^{(Alg.4)}$ the value in register R_0 at the j-th loop of Algorithm 2 and Algorithm 4, respectively. Then we see that

$$\widetilde{\eta_T}(P,Q) = \overline{\eta_T}(P,Q)^{3^{(n+1)/2}}, \tag{16}$$

since $R_{0,j}^{(Alg.2)} = (R_{0,j}^{(Alg.4)})^{3^{j+1}}$. Due to Eqs.(13) and (16), it is enough to prove that $\overline{\eta_T}(P,Q)^W = \eta_T(P,Q)^{VW}$.

The difference between Algorithm 3 and 4 causes the corresponding difference between $\eta_T(P,Q)$ and $\overline{\eta_T}(P,Q)$. There are two differences, **the first difference** is that Algorithm 3 has the program

$$\text{“ } \textbf{if } b'=1 \textbf{ then } y_p \leftarrow -y_p\text{”}, \tag{17}$$

and **the second difference** is that Algorithm 3 has the branches at Steps 1, 3, 5 and 6.

In order to investigate **the first difference**, we modify Algorithm 4 by appending the program (17) before Step 1. We call this modified algorithm as Algorithm 4', and denote by $\eta'_T(P,Q)$ the pairing value from Algorithm 4'. The relationship between $\overline{\eta_T}(P,Q)$ and $\eta'_T(P,Q)$ is obtained by Lemma 2.

Lemma 2. *We have*

$$\overline{\eta_T}(P,Q)^W = \begin{cases} (\eta'_T(P,Q)^W)^{-1} & \text{if } b'=1 \\ \eta'_T(P,Q)^W & \text{if } b'=-1 \end{cases}$$

Proof. We see that $\overline{\eta_T}$ is identical to η'_T if $b = -1$. On the other hand, $\overline{\eta_T}$ is different from η'_T if $b = 1$. We obtain $\overline{\eta_T}(P,Q) = \eta'_T(-P,Q)$ since $-P = (x_p, -y_p)$ for $P = (x_p, y_p) \in E^b(\mathbb{F}_{3^n})$. Bilinearity of $\eta_T(P,Q)^W$ products a relationship $\overline{\eta_T}(P,Q)^W = \eta'_T(-P,Q)^W = (\eta'_T(P,Q)^W)^{-1}$. □

Remark 1. $\overline{\eta_T}$ without the powering by W is not bilinear pairing. Then the powering by W is required in the statement of Lemma 2.

The second difference causes the difference between $\eta'_T(P,Q)$ and $\eta_T(P,Q)$. We soon see that $\eta'_T(P,Q) = \eta_T(P,Q)$ if $n \equiv 1 \pmod{12}$. When $n \equiv 5 \pmod{12}$, a converting $x_q \to x_q - b$, in other words $Q \to \phi^4(Q)$, in Algorithm 2 gives Algorithm 4. Then $\eta'_T(P,Q) = \eta_T(P, \phi^4(Q))$ if $n \equiv 5 \pmod{12}$. We easily see also that the relationship between $\eta'_T(P,Q)$ and $\eta_T(P,Q)$ for $n \equiv 7, 11 \pmod{12}$. Then we have

$$\eta'_T(P,Q) = \begin{cases} \eta_T(P,Q) & \text{if } n \equiv 1 \pmod{12} \\ \eta_T(P,\phi^4(Q))(= \eta_T(P,-\phi(Q))) & \text{if } n \equiv 5 \pmod{12} \\ \eta_T(P,-Q) & \text{if } n \equiv 7 \pmod{12} \\ \eta_T(P,\phi(Q)) & \text{if } n \equiv 11 \pmod{12} \end{cases} \tag{18}$$

Due to Lemma 2 and Eq.(18) we have the relationship $\overline{\eta_T}(P,Q)^W$ and $\eta_T(P,Q)^W$,

$$\overline{\eta_T}(P,Q)^W = \begin{cases} (\eta_T(P,Q)^W)^{-1} & \text{if } n \equiv 1 \pmod{12},\ b' = 1 \quad (b = 1) \\ \eta_T(P,Q)^W & \text{if } n \equiv 1 \pmod{12},\ b' = -1\ (b = -1) \\ (\eta_T(P,-\phi(Q))^W)^{-1} & \text{if } n \equiv 5 \pmod{12},\ b' = 1 \quad (b = -1) \\ \eta_T(P,-\phi(Q))^W & \text{if } n \equiv 5 \pmod{12},\ b' = -1\ (b = 1) \\ (\eta_T(P,-Q)^W)^{-1} & \text{if } n \equiv 7 \pmod{12},\ b' = 1 \quad (b = -1) \\ \eta_T(P,-Q)^W & \text{if } n \equiv 7 \pmod{12},\ b' = -1\ (b = 1) \\ (\eta_T(P,\phi(Q))^W)^{-1} & \text{if } n \equiv 11 \pmod{12},\ b' = 1 \quad (b = 1) \\ \eta_T(P,\phi(Q))^W & \text{if } n \equiv 11 \pmod{12},\ b' = -1\ (b = -1) \end{cases} \tag{19}$$

Lastly in order to show that ϕ is a homomorphism of $E^b(\mathbb{F}_{3^n})$, we show that ϕ is represented as a scalar multiplication.

Lemma 3. *For $P \in E^b(\mathbb{F}_{3^n})$, $\phi(P)$ is equal to a value in the following table.*

	$b = 1$	$b = -1$
$n \equiv 1$ *(mod 12)*	$3^n P$	$3^n P$
$n \equiv 5$ *(mod 12)*	$(-3^{(n+1)/2} + 2)P$	$(3^{(n+1)/2} + 2)P$
$n \equiv 7$ *(mod 12)*	$3^n P$	$3^n P$
$n \equiv 11$ *(mod 12)*	$(3^{(n+1)/2} + 2)P$	$(-3^{(n+1)/2} + 2)P$

Proof. See the appendix. □

Here we go back to the proof of Proposition 1. Lemma 3, Eq.(19), and the bilinearity of $\eta_T(P,Q)^W$ yield Proposition 1-(*ii*). Finally we prove Proposition 1-(*i*) in the following. V and $3^{(n+1)/2}$ are coprime to $\#E^b(\mathbb{F}_{3^n})$ with $V = \pm 1, 3^{(n+1)} \pm 2, -3^{(n+1)} \pm 2$, which means that a powering by $3^{(n+1)/2} \cdot V$ is a group isomorphism in $\mathbb{F}^*_{3^{6n}}$. The η_T^W is a non-degenerate and bilinear pairing, then the $\overline{\eta_T}^W (= \eta_T^{3^{(n+1)/2} V W})$ is also a non-degenerate and bilinear pairing.

5 Conclusion

This paper provided an explicit algorithm for computing the η_T pairing with arbitrary degree n. It has many branches based on extension degree n and the curve parameter b. Therefore, this paper also proposed the universal η_T pairing ($\widetilde{\eta_T}$ pairing) which has no branch in the program and is suitable for the efficient implementation for arbitrary extension degree n. Moreover we proved the relationship between the $\widetilde{\eta_T}$ pairing and the Tate pairing for arbitrary n.

Finally we summarize the relationship of pairings appeared in this paper in the following table.

Pairing	Properties
$\hat{e}(P,Q)$	**no branch** and **no cube root** ([11])
$\updownarrow$ Eq.(13)	
$\eta_T(P,Q)^W$	branches and cube roots (Sec.3.1)
$\updownarrow$ powering by V Proposition 1)	
$\overline{\eta_T}(P,Q)^W$	**no branch** and cube roots (Sec.4.2)
$\updownarrow$ powering by $3^{(n+1)/2}$	
$\widetilde{\eta_T}(P,Q)^W$	**no branch** and **no cube root** (Sec.3.2)

References

1. P. Barreto, "A note on efficient computation of cube roots in characteristic 3," Cryptology ePrint Archive, Report 2004/305, 2004.
2. P. Barreto, H. Kim, B. Lynn and M. Scott, "Efficient algorithms for pairing-based cryptosystems," *CRYPTO 2002*, LNCS 2442, pp.354-368, 2002.
3. P. Barreto, S. Galbraith, C. Ó hÉigeartaigh and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Designs, Codes and Cryptography*, Springer–Verlag, Vol. 42, No. 3, pp 239-271, 2007.
4. J.-L. Beuchat, M. Shirase, T. Takagi and E. Okamoto, "An algorithm for the η_T pairing calculation in characteristic three and its hardware implementation," *18th IEEE International Symposium on Computer Arithmetic, ARITH-18*, pp.97-104, 2007. (full version, Cryptology ePrint Archive, Report 2006/327, 2006.)
5. D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," *SIAM Journal of Computing*, Vol. 32, No. 3, pp. 586-615, 2003.
6. D. Boneh, C. Gentry and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," *CRYPTO 2005*, LNCS 3621, pp.258-275, 2005.
7. D. Boneh, B. Lynn and H. Shacham, "Short signature from the Weil pairing," *Journal of Cryptology*, Vol. 17, No. 4, pp. 297-319, 2004.
8. I. Duursma and H. Lee, "Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$," *ASIACRYPT 2003*, LNCS 2894, pp.111-123, 2003.
9. 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, July 2005, pp.852-860, 2005.
10. S. Galbraith, K. Harrison and D. Soldera, "Implementing the Tate pairing," *ANTS-V*, LNCS 2369, pp.324-337, 2002.

11. S. Kwon, "Efficient Tate pairing computation for supersingular elliptic curves over binary fields," Cryptology ePrint Archive, Report 2004/303, 2004.
12. V. Miller, "Short programs for functions on curves," Unpublished manuscript, 1986. available at http://crypto.stanford.edu/miller/miller.pdf .
13. MIRACL, ftp://ftp.computing.dcu.ie/pub/crypto/miracl.zip .
14. R. Ronan, C. Ó hÉigeartaigh, C. Murphy, T. Kerins and P. Barreto, "A reconfigurable processor for the cryptographic η_T pairing in characteristic 3." *Information Technology : New Generations, ITNG 2007,* pp.11-16, IEEE Computer Society, 2007.
15. M. Shirase, T. Takagi and E. Okamoto, "Some efficient algorithms for the final exponentiation of η_T pairing," *ISPEC2007*, LNCS 4464, pp.254-268, 2007.
16. J. Silverman, *The arithmetic of elliptic curves*, Springer–Verlag, 1986.

A Some Lemmas

In the appendix we prove two lemmas appeared in this paper.

Lemma 1. *Let n be an odd prime. Then*

$$g_{3^{-j}P'}(\phi(Q))^{3^j} = \begin{cases} -r_0^2 + y_p^{(-j)} y_q^{(j)} \sigma - r_0 \rho - \rho^2 & \text{if } n \equiv 1 \pmod 4, \\ -r_0^2 - y_p^{(-j)} y_q^{(j)} \sigma - r_0 \rho - \rho^2 & \text{if } n \equiv 3 \pmod 4, \end{cases}$$

for $P = (x_p, y_p),\ Q = (x_q, y_q) \in E^b(\mathbb{F}_{3^n})$, where r_0 is defined as

$$r_0 = \begin{cases} x_p + x_q + b & \text{if } n \equiv 1 \pmod 6, \\ x_p + x_q - b & \text{if } n \equiv 5 \pmod 6. \end{cases}$$

Proof. First we inspect how $P' = 3^{(n-1)/2} P$ (Eq.(14)) is represented. We see that

$$\phi^s(x,\ y) = \begin{cases} (x,\ y) & \text{if } s \equiv 0 \pmod 6, \\ (x-b,\ -y) & \text{if } s \equiv 1 \pmod 6, \\ (x+b,\ y) & \text{if } s \equiv 2 \pmod 6, \\ (x,\ -y) & \text{if } s \equiv 3 \pmod 6, \\ (x-b,\ y) & \text{if } s \equiv 4 \pmod 6, \\ (x+b,\ -y) & \text{if } s \equiv 5 \pmod 6, \end{cases} \tag{20}$$

for any $(x, y) \in E^b(\mathbb{F}_{3^n})$ and any integer s due to Eq.(5). Then we have

$$\phi^{(n-1)/2}(x,y) = \begin{cases} \phi^0(x,y) = (x,y) & \text{if } n \equiv 1 \pmod{12}, \\ \phi^2(x,y) = (x+b,y) & \text{if } n \equiv 5 \pmod{12}, \\ \phi^3(x,y) = (x,-y) & \text{if } n \equiv 7 \pmod{12}, \\ \phi^5(x,y) = (x+b,-y) & \text{if } n \equiv 11 \pmod{12}. \end{cases}$$

The notation of $a^{(i)}$ means a^{3^i}. We see that

$$\pi^n(x_p, y_p) = (x_p, y_p) \tag{21}$$

for $P = (x_p, y_p) \in E^b(\mathbb{F}_{3^n})$ since x_p and $y_p \in \mathbb{F}_{3^n}$. Then we have $\pi^{n-1}(x_p, y_p) = (x_p^{(-1)}, y_p^{(-1)})$. Therefore we see

$$P' = \begin{cases} (x_p^{(-1)},\ y_p^{(-1)}) & \text{if } n \equiv 1 \pmod{12}, \\ (x_p^{(-1)} + b,\ y_p^{(-1)}) & \text{if } n \equiv 5 \pmod{12}, \\ (x_p^{(-1)},\ -y_p^{(-1)}) & \text{if } n \equiv 7 \pmod{12}, \\ (x_p^{(-1)} + b,\ -y_p^{(-1)}) & \text{if } n \equiv 11 \pmod{12}. \end{cases}$$

Note that

$$\phi^3(x, y) = -(x, y), \tag{22}$$

due to $-(x, y) = (x, -y)$ and Eq.(20).

Next we use induction for j to prove Lemma 1. Definition equations of g_R and ψ, Eqs.(3) and (6), are utilized to prove Lemma 1 for $j = 0$.

Case of $\mathbf{n \equiv 1 \pmod{12}}$: By $P' = (x_p^{(-1)}, y_p^{(-1)})$,

$$\begin{aligned} g_{P'}(\psi(Q)) &= (y_p^{(-1)})^3 y_q \sigma - ((x_p^{(-1)})^3 - (\rho - x_q) + b)^2 \\ &= y_p y_q \sigma - (x_p + x_q + b - \rho)^2 = -r_0^2 + y_p y_q \sigma - r_0 \rho - \rho^2, \end{aligned}$$

where $r_0 = x_p + x_q + b$.

Case of $\mathbf{n \equiv 5 \pmod{12}}$: By $P' = (x_p^{(-1)} + b,\ y_p^{(-1)})$,

$$\begin{aligned} g_{P'}(\psi(Q)) &= (y_p^{(-1)})^3 y_q \sigma - ((x_p^{(-1)} + b)^3 - (\rho - x_q) + b)^2 \\ &= y_p y_q \sigma - (x_p + x_q - b - \rho)^2 = -r_0^2 + y_p y_q \sigma - r_0 \rho - \rho^2, \end{aligned}$$

where $r_0 = x_p + x_q - b$.

Case of $\mathbf{n \equiv 7 \pmod{12}}$: By $P' = (x_p^{(-1)},\ -y_p^{(-1)})$,

$$\begin{aligned} g_{P'}(\psi(Q)) &= (-y_p^{(-1)})^3 y_q \sigma - ((x_p^{(-1)})^3 - (\rho - x_q) + b)^2 \\ &= -y_p y_q \sigma - (x_p + x_q + b - \rho)^2 = -r_0^2 - y_p y_q \sigma - r_0 \rho - \rho^2, \end{aligned}$$

where $r_0 = x_p + x_q + b$.

Case of $\mathbf{n \equiv 11 \pmod{12}}$: By $P' = (x_p^{(-1)} + b,\ -y_p^{(-1)})$,

$$\begin{aligned} g_{P'}(\psi(Q)) &= (y_p^{(-1)})^3 y_q \sigma - ((x_p^{(-1)} + b)^3 - (\rho - x_q) + b)^2 \\ &= -y_p y_q \sigma - (x_p + x_q + b - \rho)^2 = -r_0^2 - y_p y_q \sigma - r_0 \rho - \rho^2, \end{aligned}$$

where $r_0 = x_p + x_q - b$.

We complete proving Lemma 1 for $j = 0$.

We suppose that Lemma 1 is held for $j = j'$. Then we easily see that Lemma 1 is also held for $j = j' + 1$ with direct computations. □

Lemma 3. *For $P \in E^b(\mathbb{F}_{3^n})$, $\phi(P)$ is equal to a value in the following table.*

	$b = 1$	$b = -1$
$n \equiv 1$ *(mod 12)*	$3^n P$	$3^n P$
$n \equiv 5$ *(mod 12)*	$(-3^{(n+1)/2} + 2)P$	$(3^{(n+1)/2} + 2)P$
$n \equiv 7$ *(mod 12)*	$3^n P$	$3^n P$
$n \equiv 11$ *(mod 12)*	$(3^{(n+1)/2} + 2)P$	$(-3^{(n+1)/2} + 2)P$

Proof. Let $P = (x_p, y_p)$ be contained in $E^b(\mathbb{F}_{3^n})$. Then we can use addition and duplication formulae of elliptic curves (see [16] for details) to obtain equations

$$\begin{cases} \pi(P) + 2P = (x_p - 1, -y_p) & \text{if } b = 1, \\ -\pi(P) + 2P = (x_p + 1, -y_p) & \text{if } b = -1. \end{cases} \tag{23}$$

The following calculations complete the proof.

Case of $\mathbf{n \equiv 1 \pmod{12}}$, $\mathbf{b = 1}$:

$$3^n P = \phi^n \pi^{2n}(P) = \phi(P) \quad \text{by (4), (20), (21)}$$

Case of $\mathbf{n \equiv 1 \pmod{12}}$, $\mathbf{b = -1}$:

$$3^n P = \phi^n \pi^{2n}(P) = \phi(P) \quad \text{by(4), (20), (21)}$$

Case of $\mathbf{n \equiv 5 \pmod{12}}$, $\mathbf{b = 1}$:

$$\begin{aligned} (-3^{(n+1)/2} + 2)P &= -\phi^{(n+1)/2}\pi^{n+1}(P) + 2P && \text{by (4)} \\ &= -\phi^3\pi(P) + 2P && \text{by (20), (21)} \\ &= \pi(P) + 2P && \text{by (22)} \\ &= \phi(P) && \text{by (23)} \end{aligned}$$

Case of $\mathbf{n \equiv 5 \pmod{12}}$, $\mathbf{b = -1}$:

$$\begin{aligned} (3^{(n+1)/2} + 2)P &= \phi^{(n+1)/2}\pi^{n+1}(P) + 2P && \text{by (4)} \\ &= \phi^3\pi(P) + 2P && \text{by (20), (21)} \\ &= -\pi(P) + 2P && \text{by (22)} \\ &= \phi(P) && \text{by (23)} \end{aligned}$$

Case of $\mathbf{n \equiv 7 \pmod{12}}$, $\mathbf{b = 1}$:

$$3^n P = \phi^n \pi^{2n}(P) = \phi(P) \quad \text{by (4), (20), (21)}$$

Case of $\mathbf{n \equiv 7 \pmod{12}}$, $\mathbf{b = -1}$:

$$3^n P = \phi^n \pi^{2n}(P) = \phi(P) \text{ by} \quad \text{(4), (20), (21)}$$

Case of $\mathbf{n \equiv 11 \pmod{12}}$, $\mathbf{b = 1}$:

$$\begin{aligned} (3^{(n+1)/2} + 2)P &= \phi^{(n+1)/2}\pi^{n+1}(P) + 2P && \text{by (4)} \\ &= \pi(P) + 2P && \text{by (20), (21)} \\ &= \phi(P) && \text{by (23)} \end{aligned}$$

Case of $\mathbf{n \equiv 11 \pmod{12}}$, $\mathbf{b = -1}$:

$$\begin{aligned} (-3^{(n+1)/2} + 2)P &= -\phi^{(n+1)/2}\pi^{n+1}(P) + 2P && \text{by (4)} \\ &= -\pi(P) + 2P && \text{by (20), (21)} \\ &= \phi(P) && \text{by (23)} \end{aligned}$$

We complete proving Lemma 3. □