Optimizing the Menezes-Okamoto-Vanstone (MOV) Algorithm for Non-Supersingular Elliptic Curves

Junji Shikata[1], Yuliang Zheng[2], Joe Suzuki[1], and Hideki Imai[3]

[1] Department of Mathematics, Graduate School of Science, Osaka University, 1-1
Machikaneyama, Toyonaka, Osaka 560-0043, Japan,
{shikata, suzuki}@math.sci.osaka-u.ac.jp,

[2] School of Comp. and Info. Tech., Monash University, McMahons Road, Frankston,
Melbourne, Victoria 3199, Australia,
yuliang@pscit.monash.edu.au,

[3] Institute of Industrial Science, University of Tokyo, 7-22-1 Roppongi, Minato-ku,
Tokyo 106-8558, Japan,
imai@iis.u-tokyo.ac.jp

Abstract. We address the Menezes-Okamoto-Vanstone (MOV) algorithm for attacking elliptic curve cryptosystems which is completed in subexponential time for supersingular elliptic curves. There exist two hurdles to clear, from an algorithmic point of view, in applying the MOV reduction to general elliptic curves: the problem of explicitly determining the minimum extension degree k such that $E[n] \subset E(F_{q^k})$ and that of efficiently finding an n-torsion point needed to evaluate the Weil pairing, where n is the order of a cyclic group of the elliptic curve discrete logarithm problem. We can find an answer to the first problem in a recent paper by Balasubramanian and Koblitz. On the other hand, the second problem is important as well, since the reduction might require exponential time even for small k. In this paper, we actually construct a novel method of efficiently finding an n-torsion point, which leads to a solution of the second problem. In addition, our contribution allows us to draw the conclusion that the MOV reduction is indeed as powerful as the Frey-Rück reduction under $n \not| q-1$, not only from the viewpoint of the minimum extension degree but also from that of the effectiveness of algorithms.

1 Introduction

1.1 History and Motivation

In 1985, Koblitz [12] and Miller [18] independently proposed the use of elliptic curves over finite fields for public-key cryptography. Since that time, elliptic curve cryptosystems have gained a tremendous amount of attention and many researchers have devoted their time to the study of elliptic curves.

The security of elliptic curve cryptosystems is based on the presumed intractability of the *Elliptic Curve Discrete Logarithm Problem*, which we abbreviate as the ECDLP. More specifically, the ECDLP can be stated as follows: Let E be an elliptic curve defined by

$$E \ : \ y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6, \ \ (a_1, a_2, a_3, a_4, a_6 \in F_q)$$

where F_q is a finite field with $q = p^m$ (p : a prime number) elements. Given a base point $P \in E(F_q)$ and $R \in \langle P \rangle$, one is asked to find an integer l such that $R = lP$, where $E(F_q)$ is the set of its F_q-rational points.

In general, thus far, it is believed that the ECDLP requires exponential time in $\log q$ to solve. Nevertheless, it has been known that, for some special cases, the ECDLP is no more difficult than the Discrete Logarithm Problem (DLP) in finite fields. Significant developments in this line of research are represented by the Menezes-Okamoto-Vanstone (MOV) algorithm [17], the Frey-Rück (FR) algorithm [9] and the Semaev-Smart-Satoh-Araki (SSSA) algorithm [24][27][21].

In the following discussion, we assume that $n = \#\langle P \rangle$, the order of a base point P, is a prime number. This condition is not restrictive, since we can reduce the composite case to the prime one by applying the Chinese Remainder Theorem and the Pohlig-Hellman algorithm.

Technically, the SSSA algorithm reduces the ECDLP to the DLP of the *additive* group structure of the base field for so-called anomalous curves and solves it in polynomial time. (For more details, see [24][27][21].) Thus, in the sequel, we will also assume that $n \not| q$, since the SSSA algorithm can be applied to the case of $n|q$.

In contrast, the MOV and FR algorithms reduce the ECDLP with the above assumptions (n prime and $n \not| q$) to the DLP in the *multiplicative* subgroup of an extension field F_{q^k} of the base field F_q and then solve the DLP using the currently known best algorithm. (For example, see [7].) A natural question that arises from an algorithmic point of view is whether it is possible to realize the reductions (i.e. transformations from the ECDLP to the DLP in finite fields) in such a way that they work efficiently.

For the FR reduction, the above question has already been answered (For example, see [10][11]): it is known that the FR reduction can work in probabilistic polynomial time in $k \log q$. Here k is explicitly given as the smallest positive integer with $q^k \equiv 1 \bmod n$. (Note that this condition follows from the requirement that F_{q^k} must contain n-th roots of unity.) Thus, if such a k is small enough to solve the DLP in $F_{q^k}^*$ in subexponential time in $\log q$, the reduction itself is always completed in polynomial time in $\log q$. Consequently, in such a case, the FR algorithm is completed in subexponential time in $\log q$. In particular, if $n|q-1$, we have no need to extend the base field F_q and the FR reduction can be easily applied.

On the other hand, for the MOV reduction, the above question has not been explicitly answered yet: thus far, it is well known that, for supersingular curves,

1. the necessary minimum extension degree k is at most six; and

2. the MOV reduction (transformation from ECDLP to DLP in a finite field) is completed in probabilistic polynomial time in $k \log q$ ($k \leq 6$),

and so the MOV algorithm for supersingular curves is completed in subexponential time in $\log q$. However, there exist two major problems that occur as obstacles, from an algorithmic point of view, in applying the MOV reduction to general elliptic curves (assuming that $n = \#\langle P \rangle$ is prime number and $n \not| q$):

1. the problem of explicitly determining the smallest positive integer k such that $E[n] \subset E(F_{q^k})$, where $E[n]$ is the set of n-torsion points.
2. the problem of efficiently finding an n-torsion point Q such that $e_n(P, Q)$ has order n (i.e. $e_n(P, Q) \neq 1$ because of the assumption that n is a prime number.), where e_n is the Weil pairing. (In the sequel, we refer to such an n-torsion point Q as a "good" n-torsion point.)

For the first problem, we can find an answer to it in a paper by Balasubramanian and Koblitz [3]. They proved that if $n \not| q - 1$, k is the smallest positive integer such that $q^k \equiv 1 \bmod n$. (It is interesting to note that this condition is identical to the one under which the FR reduction is applied.) In the same paper, they also suggest that we need $k = n$ if $n|q - 1$ and $E[n] \not\subset E(F_q)$. Thus, when n is much larger than $\log q$, we may give up applying the MOV algorithm since the extension degree in this case is too large in order for the reduced DLP in $F^*_{q^k}$ to be solved in subexponential time in $\log q$.

For the second problem, we cannot find any answer which covers all the case: a simple and widely well-known method generally requires exponential time in $\log q$ even if k is small (Section 4.1). Moreover, the methods using the multiplication by constant maps in a suitable way might also take exponential time in $\log q$ for the general case (Section 4.2).

Thus, in order to reach the valid conclusion that the MOV algorithm is always completed in subexponential time in $\log q$ if the DLP in $F^*_{q^k}$ is solved in subexponential time in $\log q$, an efficient method which solve the second problem above will be desired.

1.2 Main Result

The major contribution of this paper is to solve the second problem described earlier by constructing a novel method which finds a "good" torsion point required in evaluating the Weil pairing in probabilistic polynomial time in $k \log q$, under the most reasonable assumptions stated above (i.e. n is a prime such that $n \not| q$ and $n \not| q-1$). This expected running time is optimal, since it always means probabilistic polynomial time in $\log q$ whenever k is small enough to solve the DLP in $F^*_{q^k}$ in subexponential time in $\log q$. As a result, we obtain an optimized MOV algorithm for general elliptic curves.

The key idea which leads us to successfully finding an n-torsion point efficiently is to construct a homomorphism $f : E(F_{q^k}) \rightarrow E(F_{q^k})$ such that $Imf = E[n]$. We will see that it is generally possible by using the q-th power Frobenius map under a certain condition.

Now, we turn our attention to comparing the MOV and FR reductions. It may have been believed by some cryptographers that assuming $n \not| q-1$, the MOV reduction is as powerful as the FR reduction in the sense that their minimum extension degrees k coincide when the base field F_q is extended to F_{q^k} in order to apply those reductions. However, so far there has been a lack of a formal proof that supports the belief. As pointed out in [11], the problem of efficiently finding an n-torsion point required in evaluating the Weil pairing should be solved as well. Thus, our contribution allows us to finally draw the conclusion that the MOV reduction is indeed as powerful as the FR reduction under $n \not| q-1$, in a true sense: not only from the viewpoint of the minimum extension degree of the base field but also from that of the effectiveness of algorithms.

The rest of this paper is organized as follows: In Section 2, we briefly review some basic facts on elliptic curves over finite fields and the MOV algorithm. In Section 3, we consider the problem of determining the minimum extension degrees explicitly and describe the answer to it obtained by Balasubramanian and Koblitz. In Section 4, we consider the problem of finding a "good" n-torsion point efficiently. Three different methods are considered to solve the problem. The third method is completed in probabilistic polynomial time in $k \log q$ for the general case $n \not| q-1$. Finally, based on the efficient method in the previous section, in Section 5 we actually realize an optimized MOV algorithm for general elliptic curves under $n \not| q-1$ and estimate its running time.

2 Preliminaries

In this section, we briefly review some materials on elliptic curves over finite fields. (See [25] for more details.)

Let F_q be a finite field with $q = p^m$ elements, where p is a prime number, and $\bar{F_q}$ its algebraic closure. Let E be an elliptic curve over F_q given by the Weierstrass equation

$$E: \; y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

whose coefficients lie in F_q. For each extension field K of F_q, $E(K)$ is given by

$$E(K) = \{(x, y) \in K \times K | (x, y) \text{ satisfies } \; (1) \; \} \cup \{O\}$$

where O is a special point, called the point at infinity. There is an abelian group structure on the points of $E(K)$, in which O serves as its identity element, given by the so-called *tangent-and-chord* method. We express its abelian structure additively.

Let n be a positive integer relatively prime to p, the characteristic of F_q. The *Weil pairing* is a map

$$e_n : \; E[n] \times E[n] \longrightarrow \mu_n \subset \bar{F_q}$$

where $E[n] = \{T \in E(\bar{F_q}) | [n]T = O\}$ is the group of n-torsion points and μ_n is the subgroup of n-th roots of unity in $\bar{F_q}$. For properties of the Weil pairing, see [25] [16].

Let $P \in E[n]$ be a point of order n. Then, we have the following:

Proposition 1. ([25][17]) *There exists $Q \in E[n]$ such that $e_n(P,Q)$ is a primitive n-th root of unity. Therefore,*

$$f_Q : \langle P \rangle \longrightarrow \mu_n, \quad f_Q(S) = e_n(S,Q)$$

is a group isomorphism.

Based on this fact, the framework of the MOV algorithm can be described as follows:

Algorithm 1 ([16] [17])

Input: *An element $P \in E(F_q)$ of order n, and $R \in \langle P \rangle$.*
Output: *An integer l such that $R = [l]P$.*
Step 1: *Determine the minimum positive integer k such that $E[n] \subset E(F_{q^k})$.*
Step 2: *Find $Q \in E[n]$ such that $\alpha = e_n(P,Q)$ has order n.*
Step 3: *Compute $\beta = e_n(R,Q)$.*
Step 4: *Compute l, the discrete logarithm of β to the base α in $F^*_{q^k}$.*

This algorithm is somewhat incomplete in that the methods for determining k and for finding a point Q are not explicitly given. For supersingular elliptic curves, the methods which settle those problems are given in [17]; the resulting minimum k are $k = 1, 2, 3, 4$, or 6, and for each corresponding k, Q is efficiently obtained by using the group structure of $E(F_{q^k})$. Therefore, for supersingular elliptic curves, the reduction is completed in probabilistic polynomial time in $\log q$ and the algorithm mentioned above takes probabilistic subexponential time in $\log q$.

In the following sections, we consider the two problems described in Section 1 not only for the supersingular case but also for the non-supersingular (ordinary) case .

3 Determining the Minimum Extension Degrees

In this section, we consider the problem of determining the minimum positive integer k such that $E[n] \subset E(F_{q^k})$.

The following proposition is proved by R. Schoof [22].

Proposition 2. ([22]) *Let p be the characteristic of F_{q^k}, n a natural number with $p \not| n$ and t_k denote the trace of the q^k-th power Frobenius map ϕ of E. The following are equivalent;*

(1) $E[n] \subset E(F_{q^k})$

(2) $n^2 | \#E(F_{q^k}),\ n | q^k - 1$ *and either* $\phi \in Z$ *or* $\mathcal{O}(\frac{{t_k}^2 - 4q^k}{n^2}) \subset End_{F_{q^k}}(E)$

where $\mathcal{O}(\frac{{t_k}^2-4q^k}{n^2})$ is an order of discriminant $\frac{{t_k}^2-4q^k}{n^2}$.

However, from an algorithmic point of view, a more explicit form of k is needed to realize the MOV reduction. With the assumption that n is a prime number such that $n|\#E(F_q)$ and $n \not| q-1$, Balasubramanian and Koblitz [3] have obtained the following result:

Proposition 3. ([3]) *Let E be an elliptic curve defined over F_q, and suppose that n is a prime number such that $n|\#E(F_q)$, $n \not| q-1$. Then, $E[n] \subset E(F_{q^k})$ if and only if $n|q^k-1$.*

Remark 1. It is important to note that Balasubramanian and Koblitz's results also suggest $k = n$ if $n|q-1$ and $E[n] \not\subset E(F_q)$. Thus, in this case, when n is much larger than $\log q$, we may give up applying the MOV algorithm since the extension degree in this case is too large in order for the reduced DLP in $F_{q^k}^*$ to be solved in subexponential time in $\log q$.

4 Three Methods for Finding n-Torsion Points

In this section, we consider the problem of finding an n-torsion point $Q \in E[n]$ such that $\alpha = e_n(P,Q)$ has order n. (See Algorithm 1 in Section 2.) We refer to such an n-torsion point Q as a "good" n-torsion point.

As before, we assume the following:

Assumption 1 *(1) n is a prime number; (2) $n \not| q$; (3) $n \not| q-1$.*

The first condition is not restrictive, since we can reduce the composite case to the prime one by applying the Chinese Remainder Theorem and the Pohlig-Hellman algorithm; the second one is necessary, since the Weil pairing is not defined otherwise; the third one is reasonable from the result by Balasubramanian and Koblitz. (See Remark in Section 3.)

Also, as before, we use the following notation: P is a base point of order n. (Thus, $E(F_q)[n] = \langle P \rangle \cong Z/n$.); k is the minimum positive integer such that $E[n] \subset E(F_{q^k})$, or equivalently k is the minimum positive integer with $n|q^k-1$. (See Proposition 3 in Section 3.)

Let N_k be the number of F_{q^k}-rational points on E, and $E(F_{q^k})_n$ the n-primary part of $E(F_{q^k})$, i.e.

$$N_k = \#E(F_{q^k}), \qquad E(F_{q^k})_n = \bigcup_{i \geq 1} E(F_{q^k})[n^i],$$

and, let $d = v_n(N_k)$ denote the largest integer such that $n^d|N_k$.

Now, we provide three different methods to find a "good" n-torsion point; the first one, which is considerable simple, repeatedly chooses $Q \in E(F_{q^k})$ until both $Q \in E[n]$ and $e_n(P,Q) \neq 1$ are satisfied; the second one is a method using the multiplication by constant maps and can be regarded as a generalized version of the algorithm that Menezes, Okamoto, and Vanstone considered in the original paper [17] on the MOV reduction for supersingular elliptic curves; and the third one, which is constructed based on Theorem 1 given later, can be applied to the general case $n \not| q-1$ and is completed in probabilistic polynomial time in $k \log q$. It turns out that the second one takes a smaller expected number of iterations than the first one in order to obtain a "good" n-torsion point. However, they generally require exponential time in $\log q$. The third one, our final goal of this section, is optimal, since it is completed in probabilistic polynomial time in $k \log q$ for general elliptic curves.

4.1 A Simple Method

We assume that $E(F_{q^k}) \cong Z/cd_1n \times Z/cn$ $(cn|q^k-1)$. (Note that the group structure of $E(F_{q^k})$ can be always expressed in this form [16].)

The first method is simple:

Procedure 1

Step 1: *Choose* $Q \in E(F_{q^k})$ *randomly.*
Step 2: *Check if* $Q \in E[n]$ *by computing* $[n]Q$*. If* $Q \notin E[n]$*, go to Step 1.*
Step 3: *Compute* $\alpha = e_n(P,Q)$*. If* $\alpha = 1$*, go to Step 1.*

In *Step 1*, we first pick an element $x = a$ in F_{q^k} to substitute it to the equation (1). Then we check if the quadratic equation with respect to y has a solution in F_{q^k}. If it does, we solve the quadratic equation in a usual manner. (See, for example, [13][16] for the details.) Also, for *Step 3*, there is a standard procedure to compute the Weil pairing. (See, for example, [16].) Note that we can execute this method even if the group structure of $E(F_{q^k})$ is unknown.

If Procedure 1 is applied, the probability of finding a "good" point Q for each iteration is

$$\frac{\#E[n]}{\#E(F_{q^k})} \times \frac{\#E[n] - \#\langle P\rangle}{\#E[n]} = \frac{n^2}{d_1c^2n^2} \times \frac{n^2-n}{n^2} = \frac{1}{d_1c^2}(1-\frac{1}{n}).$$

Thus, the success probability for each iteration is approximately $\frac{1}{d_1c^2}$ since n is assumed to be large enough. If $n = O(q)$, the expected number of iterations is approximately $d_1c^2 = N_k/n^2 = O(q^{k-2})$, where for the last equality $n = O(q)$ and the Hasse bound [25] have been applied. Therefore, if $k > 2$, the above method is no longer efficient, since it takes exponential time in $\log q$.

4.2 Methods Using the Multiplication by Constant Maps

The second method is a generalized version of that considered in the original paper [17] on the MOV reduction for supersingular elliptic curves. Two versions of this method are considered. One may use one of these versions depending on whether the knowledge of the group structure of $E(F_{q^k})$ is required or not (Procedure 2 and 3, respectively). Procedure 2 is considered in [11]. However, Procedure 3 is different from the method in [11], since our method does not need the information of the complete group structure of $E(F_{q^k})$.

We first consider the case that the group structure $E(F_{q^k}) \cong Z/cd_1n \times Z/cn$ $(cn|q^k-1)$ is known.

Procedure 2

Step 1: *Set* $v_n(d_1)$ *the largest integer such that* $n^{v_n(d_1)}|d_1$ *and set* $d_2 := d_1/n^{v_n(d_1)}$.
Step 2: *Choose* $Q \in E(F_{q^k})$ *randomly.*
Step 3: *Set* $Q' = [cd_2]Q \in E[n^{v_n(d_1)+1}] \cap E(F_{q^k}) \cong Z/n^{v_n(d_1)+1} \times Z/n$.
Step 4: *Check if* $Q' \in E[n]$ *by computing* $[n]Q'$. *If* $Q' \notin E[n]$, *go to Step 2.*
Step 5: *Compute* $\alpha = e_n(P, Q')$. *If* $\alpha = 1$, *go to Step 2.*

If Procedure 2 is applied, the probability of finding a "good" point Q' for each iteration is

$$\frac{\#E[n]}{\#(E[n^{v_n(d_1)+1}] \cap E(F_{q^k})} \times \frac{\#E[n] - \#\langle P\rangle}{\#E[n]} = \frac{n^2}{n^{v_n(d_1)+2}} \times \frac{n^2-n}{n^2} = \frac{1}{n^{v_n(d_1)}}(1-\frac{1}{n})$$

In particular, if $n \not| d_1$, this method is simplified to:

Procedure 2'

Step 1: *Choose* $Q \in E(F_{q^k})$.
Step 2: *Set* $Q' = [cd_1]Q \in E[n]$.
Step 3: *Compute* $\alpha = e_n(P, Q')$. *If* $\alpha = 1$, *go to Step 1.*

The probability of finding a "good" point Q' for each iteration is $1 - 1/n$. Note that, for supersingular elliptic curves, $d_1 = 1$ and this method coincides with what was used in [17]. In this sense, Procedure 2 can be regarded as a generalized version of that used in [17].

The probability of finding a "good" point for each iteration in Procedure 2 is approximately $1/n^{v_n(d_1)}$, and the expected number of iterations is approximately $n^{v_n(d_1)}$, which is smaller than that of Procedure 1 since $n^{v_n(d_1)} \leq d_1 \leq d_1c^2$.

We next consider the case that the group structure of $E(F_{q^k})$ is unknown beforehand. For finding the group structure of $E(F_{q^k})$, we apply Miller's algorithm, which finds the pair (n_1, n_2) such that $E(F_{q^k}) \cong Z/n_1 \times Z/n_2$ $(n_2|n_1, n_2|q^k - 1)$ assuming the knowledge of the factorization of N_k. (For the details of Miller's algorithm, see [16] [17].) However, since we are looking at the n-primary part, all we need is the information on the group structure of that, i.e. the pair (r, s) such that $E(F_{q^k})_n \cong Z/n^r \times Z/n^s$ $(1 \leq s \leq r)$. Then, we can avoid computing the factorization of N_k, and consequently that leads to a great saving of computation. Thus, in the following procedure (Procedure 3), we make use of a simplified version of Miller's algorithm (*N_Miller*) which computes the group structure of the n-primary part without the knowledge of the factorization of N_k. The essential difference between Procedures 2 and 3 lies in this point.

Procedure 3

Step 1: *Compute* $N_1 = \#E(F_q)$.
Step 2: *Compute* $N_k = \#E(F_{q^k})$ *from* $N_1 = \#E(F_q)$, *using the Weil Theorem, and* $d = v_n(N_k)$.
Step 3: *Execute N_Primary_Miller to get the pair* $(r,\ s)$.
Step 4: *Compute* $t = N_k/n^{r+1}$.

Step 5: *Choose* $Q \in E(F_{q^k})$ *randomly, and compute* $Q' = [t]Q$.
Step 6: *Check if* $Q' \in E[n]$ *by computing* $[n]Q'$. *If* $Q' \notin E[n]$, *go to Step 5.*
Step 7: *Compute* $\alpha = e_n(P, Q')$. *If* $\alpha = 1$, *go to Step 5.*

N_Miller :

1) *Pick* $V, W \in E(F_{q^k})$ *randomly.*
2) *Compute* $V' = [N_k/n^d]V$ *and* $W' = [N_k/n^d]W$.
3) *Compute* $ord(V')$, $ord(W')$ *(the orders of* V', W', *respectively),*
and set $r = max\{v_n(ord(V')), v_n(ord(W'))\}$.
4) *Compute* $\delta = e_{n^r}(V', W')$ *and its order* $n^s = ord(\delta)$.
5) *If* $r + s = d$, *then return (r, s). Otherwise, go to 1).*

We provide explanation of each step in the above method.

In *Step 1,* we compute N_1 in polynomial time, using the Schoof-Elkies-Atkin algorithm and its variants [23][5][14][6][1][2][8][20].

As described earlier, *N_Miller* is regarded as Miller's algorithm that finds the group structure of the n-primary part $E(F_{q^k})_n$ of $E(F_{q^k})$. In 2), note that the multiplication by N_k/n^d map $[N_k/n^d] : E(F_{q^k}) \rightarrow E(F_{q^k})$ is an abelian group homomorphism and hence it preserves the uniform distribution. Moreover, since its image is the n-primary part $E(F_{q^k})_n$, we can obtain V', $W' \in E(F_{q^k})_n$ randomly if we pick V, $W \in E(F_{q^k})$ randomly. In 5), if $r + s = d$, we can see that the group structure of the n-primary part is isomorphic to $Z/n^r \times Z/n^s$, and also the probability of success is

$$\frac{\varphi(n^r)\varphi(n^s)}{n^{r+s}} = \frac{n^{r-1}(n-1)n^{s-1}(n-1)}{n^{r+s}} = (1 - \frac{1}{n})^2,$$

where φ is the Euler function.

In *Steps 4* and *5,* we note that $t = \frac{N_k}{n^{r+1}} = \frac{N_k}{n^d} \cdot n^{s-1}$. Therefore, the image of the multiplication by t map $[t] = [n^{s-1}] \circ [N_k/n^d] : E(F_{q^k}) \longrightarrow E(F_{q^k})$ is exactly isomorphic to $Z/n^{r-s+1} \times Z/n$. Thus $[t]$ is an analogue of $[cd_2]$ in *Step 3* of Procedure 2, although $[t]$ is not correctly corresponding to $[cd_2]$.

Finally, we briefly analyze the time complexity of our method. (i.e. Procedure 2 and 3.) From the considerations described earlier, it follows that the success probability for each iteration is approximately $1/n^{v_n(d_1)}$ and the expected number of iterations is $O(n^{v_n(d_1)})$. Thus, if $v_n(d_1) = 0$, i.e. $d = v_n(N_k)$ is even and the group structure of the n-primary part $E(F_{q^k})_n$ is isomorphic to $Z/n^{\frac{d}{2}} \times Z/n^{\frac{d}{2}}$, they are completed in probabilistic polynomial time in $k \log q$. Otherwise, they are no longer efficient, since it takes exponential time in $\log q$ for large n.

Remark 2. It is possible to improve the method in order to make the success probability high. For example, after picking a point $Q \in E(F_{q^k})_n$ randomly by using the map $[N_k/n^d]$, we compute its order, say, n^l. Then we can obtain $Q' = [n^{l-1}]Q \in E[n]$. This might be familiar to some people. The success probability of this method is better than that of the above methods. However, in general cases, this procedure also requires the expected number of iterations $O(n^{v_n(d_1)})$ and the time complexity remains same. More precisely, we cannot reduce the expected running time when $\langle P\rangle = \langle n^{r-1}S\rangle$, where $E(F_{q^k})_n \cong Z/n^r \times Z/n^s$ $(1 \le s < r)$ and S is an element of order n^r. Otherwise, the expected number of iterations is almost one. In fact, this is clear by considering the two cases: $E(F_{q^k})_n \cong Z/n^sZ \times Z/n^sZ$; $E(F_{q^k})_n \cong Z/n^rZ \times Z/n^sZ$ $(1 \le s < r)$ with $\langle P\rangle \neq \langle n^{r-1}S\rangle$. The first case has been already considered in the previous methods while the second case will be addressed in the following section. The above case (i.e. $\langle P\rangle = \langle n^{r-1}S\rangle$) is essentially solved by proposing the next method in Section 4.3.

4.3 An Efficient Method for General Elliptic Curves.

The methods described before are not always completed in polynomial time in $k \log q$ for the general case. In other words, we need some assumptions in order for them to be completed in polynomial time in $k \log q$. Thus, we consider to remove this restriction in this subsection. The key idea is to construct a homomorphism $f : E(F_{q^k}) \to E(F_{q^k})$ such that $\mathrm{Im} f = E[n]$, and we will see that it is possible by using the q-th power Frobenius map ϕ under a certain condition.

As a natural situation, we assume that the group structure of $E(F_{q^k})$ is unknown beforehand. The following is our proposed method for general elliptic curves.

Procedure 4

Step 1: *Compute* $N_1 = \#E(F_q)$.
Step 2: *Compute* $N_k = \#E(F_{q^k})$ *from* N_1, *and* $d = v_n(N_k)$.
Step 3: *Execute N_Miller to obtain the pair* (r, s). *If* $r = s$, *go to Step 5.*
Step 4:
  (4-1) *Choose* $Q \in E(F_{q^k})$ *randomly.*
  (4-2) *Compute* $Q' = [N_k/n^{s+1}]Q$. *If* $Q' = O$, *go to (4-1). Otherwise, compute* $Q'' = (\phi - 1)Q'$.
  (4-3) *If* $Q'' \neq O$, *compute* $\alpha = e_n(P, Q')$ *and go to Step 6.*
Step 5:
  (5-1) *Choose* $Q \in E(F_{q^k})$ *randomly.*
  (5-2) *Compute* $Q' = (\phi - 1)^{r-s} \circ [N_k/n^{r+1}]Q$. *(We define* $(\phi - 1)^0 :=$ *id: identity map.)*
  (5-3) *Compute* $\alpha = e_n(P, Q')$. *If* $\alpha = 1$, *go to (5-1).*
Step 6: *Store* Q' *and* α.

We provide explanation of each step in the above method.

In *Step 3*, we can know the group structure of the n-primary part $E(F_{q^k})_n \cong Z/n^rZ \times Z/n^sZ$ $(1 \le s \le r)$. If $r = s$, the rest of this method is same as Procedure 3, which is proposed in Section 4.2.

In *Step 4*, we assume that $E(F_{q^k})_n = \langle S\rangle \times \langle T\rangle \cong Z/n^rZ \times Z/n^sZ$ $(1 \le s < r)$, where S and T are generators of orders n^r and n^s, respectively. (However, note that we cannot now find such a basis $\{S, T\}$ from an algorithmic point of view.) The image of the multiplication by $N_k/n^{s+1} = n^{r-1} \times N_k/n^d$ map $[n^{r-1}] \circ [N_k/n^d] : E(F_{q^k}) \to E(F_{q^k})$ is isomorphic to Z/nZ. We can know whether $\langle P\rangle = \langle n^{r-1}S\rangle$ or not by checking whether $Q'' = O$ or not. In fact, since the order of Q' is n and $Q'(\neq O) \in \langle n^{r-1}S \rangle$, and since we have assumed $E(F_q)[n] = \langle P \rangle$, it follows that $\langle P \rangle = \langle n^{r-1}S \rangle \Leftrightarrow Q' \in \langle P \rangle \Leftrightarrow (\phi - 1)Q' = O$. In order to check whether $\langle P \rangle = \langle n^{r-1}S \rangle$, we need $Q' \neq O$, and its success probability is

$$\frac{\varphi(n^r)n^s}{n^{r+s}} = \frac{n^{r-1}(n-1)n^s}{n^{r+s}} = 1 - \frac{1}{n}$$

if we choose $Q \in E(\boldsymbol{F}_{q^k})$ randomly. When $\langle P \rangle \neq \langle n^{r-1}S \rangle$, we can obtain $\alpha = e_n(P, Q') \neq 1$.

In *Step 5*, we already know that $E(F_{q^k})_n \cong Z/n^sZ \times Z/n^sZ$, or $E(F_{q^k})_n = \langle S \rangle \times \langle T \rangle \cong Z/n^rZ \times Z/n^sZ \quad (1 \leq s < r)$ with $\langle P \rangle = \langle n^{r-1}S \rangle$. For the first case, the rest of the method is same as Procedure 3. Therefore, when $r = s$, the success probability for each iteration is $1 - 1/n$. For the second case, in order to explain the validity of this step, we need the following theorem:

Theorem 1. *We assume that*

$$\begin{aligned} E(F_{q^k})_n &= \langle S \rangle \times \langle T \rangle \cong Z/n^rZ \times Z/n^sZ \quad (1 \leq s < r), \\ \langle P \rangle &= \langle n^{r-1}S \rangle, \end{aligned}$$

where S and T are generators of orders n^r and n^s, respectively. Consider the map

$$f = (\phi - 1)^{r-s} \circ [n^{s-1}] \circ [N_k/n^d] : \ E(F_{q^k}) \longrightarrow E(F_{q^k}).$$

Then we have

$$Imf \cong Z/nZ \times Z/nZ.$$

Proof. Consider the multiplication by N_k/n^d map $[N_k/n^d] : \ E(F_{q^k}) \longrightarrow E(F_{q^k})$. It is easy to see that its image is the n-primary part $E(F_{q^k})_n$ of $E(F_{q^k})$. We define $f^{(r-s)} := (\phi - 1)^{r-s} : E(F_{q^k}) \longrightarrow E(F_{q^k})$, where ϕ is q-th power Frobenius map. Then, from Lemma 1, which will be given below, it follows that $\mathrm{Im}(f^{(r-s)} \circ [N_k/n^d]) = \mathrm{Im}(f^{(r-s)}|_{E(F_{q^k})_n}) \cong Z/n^sZ \times Z/n^sZ$. Moreover, by composing the map $[n^{s-1}]$ with it, we have $\mathrm{Im}([n^{s-1}] \circ f^{(r-s)} \circ [N_k/n^d]) \cong Z/nZ \times Z/nZ$. Therefore, $\mathrm{Im}f \cong Z/nZ \times Z/nZ$ follows, since

$$[n^{s-1}] \circ f^{(r-s)} \circ [N_k/n^d] = f^{(r-s)} \circ [n^{s-1}] \circ [N_k/n^d]. \qquad \square$$

Lemma 1. *We assume that*

$$E(F_{q^k})_n = \langle S \rangle \times \langle T \rangle \cong Z/n^rZ \times Z/n^sZ \quad (1 \leq s < r),$$

where S and T are generators of orders n^r and n^s, respectively, and

$$\langle P \rangle = \langle n^{r-1}S \rangle.$$

We consider the map :

$$f^{(i)} = (\phi - 1)^i : \ E(F_{q^k}) \longrightarrow E(F_{q^k}) \ \ (0 \leq i \leq r - s),$$

where ϕ is the q-th power Frobenius map. Then we have

$$Im(f^{(i)}|_{E(F_{q^k})_n}) \cong Z/n^{r-i}Z \times Z/n^s Z$$

and $\mathrm{Im}(f^{(i)}|_{E(F_{q^k})_n})$ *is generated by $f^{(i)}(S)$ and $f^{(i)}(T)$ of orders n^{r-i} and n^s, respectively.*

Proof. The proof is given in Appendix A. □

From Theorem 1, it follows that $Q' \in E[n]$ in *Step (5-2)*, since $(\phi - 1)^{r-s} \circ [N_k/n^{r+1}] = (\phi - 1)^{r-s} \circ [n^{s-1}] \circ [N_k/n^d]$. Moreover, when $r > s$, the effectiveness of *Step 5* is justified by the following proposition:

Proposition 4. *We assume that*

$$\begin{aligned} E(F_{q^k})_n &= \langle S \rangle \times \langle T \rangle \cong Z/n^r Z \times Z/n^s Z \ \ (1 \leq s < r) \\ \langle P \rangle &= \langle n^{r-1} S \rangle, \end{aligned}$$

where S and T are generators of orders n^r and n^s, respectively. Then, in Step 5 of Procedure 4, the probability of obtaining $\alpha \neq 1$ is $1 - 1/n$.

Proof. Since $\frac{N_k}{n^{r+1}} = n^{s-1} \cdot \frac{N_K}{n^d}$ $(d = v_n(N_k) = r + s)$, we have

$$f = (\phi - 1)^{r-s} \circ [N_k/n^{r+1}] = (\phi - 1)^{r-s} \circ [n^{s-1}] \circ [N_k/n^d].$$

The map $f : E(F_{q^k}) \longrightarrow E(F_{q^k})$ is an abelian group homomorphism and it preserves the uniform distribution. Moreover, its image is isomorphic to $Z/nZ \times Z/nZ$. (See Theorem 1.) Thus the probability of finding $Q \in E(F_{q^k})$ such that $e_n(P, f(Q)) \neq 1$ is

$$\frac{\#E[n] - \#\langle P \rangle}{\#E[n]} = \frac{n^2 - n}{n^2} = 1 - \frac{1}{n}. \qquad \Box$$

Finally, from the considerations above, it follows that the probability of success in Procedure 4 is approximately one and that it is completed in probabilistic polynomial time in $k \log q$. (See Section 5.2. for more details.)

5 Optimizing the MOV Algorithm for General Elliptic Curves

In this section, we actually realize the MOV algorithm for general elliptic curves under Assumption 1, using the results obtained in the previous sections.

5.1 Description of an Optimized MOV Algorithm

The MOV algorithm is completed, based on the results in previous sections, as follows:

Algorithm 2 **(An Optimized MOV Algorithm)**

```
Input: An elliptic curve $E$, a base point $P \in E(F_q)$ and $R \in \langle P \rangle$.
Output: An integer $l$ such that $R = [l]P$.
Step 1: Determine the minimum positive integer $k$ such that $q^k \equiv 1 \mod n$.
Step 2:
    (2-1) Compute $N_1 = \#E(F_q)$.
    (2-2) Compute $N_k = \#E(F_{q^k})$ from $N_1$, using the Weil Theorem, and $d = v_n(N_k)$.
    (2-3) Execute N_Miller to obtain the pair $(r,\ s)$. If $r = s$, go to (2-5).
    (2-4) :
        (2-4-1) Choose $Q \in E(F_{q^k})$ randomly.
        (2-4-2) Compute $Q' = [N_k/n^{s+1}]Q$. If $Q' = O$, go to (2-4-1). Otherwise, compute $Q'' = (\phi - 1)Q'$.
        (2-4-3) If $Q'' \neq O$, compute $\alpha = e_n(P, Q')$ and go to (2-6).
    (2-5) :
        (2-5-1) Choose $Q \in E(F_{q^k})$ randomly.
        (2-5-2) Compute $Q' = (\phi - 1)^{r-s} \circ [N_k/n^{r+1}]Q$.
        (2-5-3) Compute $\alpha = e_n(P, Q')$. If $\alpha = 1$, go to (2-5- 1).
    (2-6) Store $Q'$ and $\alpha$.
Step 3: Compute $\beta = e_n(R, Q')$.
Step 4: Compute $l$, the discrete logarithm of $\beta$ to the base $\alpha$ in $F^*_{q^k}$.

    N_Miller :

1) Pick $V, W \in E(F_{q^k})$ randomly.
2) Compute $V' = [N_k/n^d]V$ and $W' = [N_k/n^d]W$.
3) Compute $ord(V')$, $ord(W')$ and set $r = max\{v_n(ord(V')), v_n(ord(W'))\}$.
4) Compute $\delta = e_{n^r}(V', W')$ and its order $n^s = ord(\delta)$.
5) If $r + s = d$, then return $(r,\ s)$. Otherwise, go to 1).
```

5.2 Success Probability and Running Time

We consider the success probability and running time of Algorithm 2.

Let k be the minimum positive integer with $q^k \equiv 1 \bmod n$.

- Success Probability:
  1. the success probability in *N_Miller* is approximately $(1 - 1/n)^2$. (See Section 4.2.)
  2. the success probability in *Step (2-4)* is approximately $1 - 1/n$. (See Section 4.3.)
  3. the success probability in *Step (2-5)* is approximately $1 - 1/n$. (See Section 4.3.)

Therefore, once we determine the value k in *Step 1*, the success probability of the rest of the reduction (i.e. *Step 2 and Step 3*) is approximately one.

- Running Time:
  We assume that the usual multiplication algorithms are used, so that multiplying two elements of length N takes time $O(N^2)$. We estimate the running time of the following major computation.
  1. Computation of $\#E(F_q)$ using the Schoof-Elkies-Atkin algorithm and its variants (in *Step (2-1)*): this procedure requires $O(\log^6 q)$.
  2. Picking a random point on $E(F_{q^k})$: this procedure requires $O(k^3 \log^3 q)$.
  3. Computation of Q' (and V', W'): computation of Q' in *Step (2-4-2)* requires $O((\log N_k)(k \log q)^2) = O((k \log q)(k \log q)^2) = O(k^3 \log^3 q)$, where $N_k = \#E(F_{q^k})$. Similarly, computation of V' and W' requires $O(k^3 \log^3 q)$. Computation of Q' in *Step (2-5-2)* requires $O(k^3 \log^3 q + (r - s)(\log q)(k \log q)^2) = O(k^3 \log^3 q)$.
  4. Computation of the Weil pairing $e_n(P, Q')$: this procedure requires $O(k^3 \log^3 q + (\log n)(k \log q)^2) = O(k^3 \log^3 q + k^2 \log^3 q) = O(k^3 \log^3 q)$. Similarly, computation of the Weil pairing $e_{n^r}(V', W')$ in *N_Miller* requires $O(k^3 \log^3 q + (r \log n)(k \log q)^2) = O(k^3 \log^3 q)$.

  Also, each procedure in *Step 2* and *Step 3*, except the above, require at most $O(k^3 \log^3 q)$. Therefore, once we determine the value k in *Step 1*, the rest of the reduction (i.e. *Step 2* and *Step 3*) is completed in polynomial time in $k \log q$, more precisely, $O(k^3 \log^3 q + \log^6 q)$.

Acknowledgements

The authors would like to thank Nigel Smart for useful comments on the earlier version of this paper, Shigenori Uchiyama and Taiichi Saito for fruitful discussion at some domestic conferences and the referees for useful comments.

References

1. A. O. L. Atkin, "The number of points on an elliptic curve modulo a prime", Draft, 1988.
2. A. O. L. Atkin, "The number of points on an elliptic curve modulo a prime (ii)," Draft, 1992.
3. R. Balasubramanian and N. Koblitz, "The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm," in Journal of Cryptology 11, pp.141-145, 1998.
4. H. Cohen, "A Course in Computational Algebraic Number Theory," Springer-Verlag, Berlin, 1993.
5. J. M. Couveignes and F. Morain, "Schoof's algorithm and isogeny cycles," in the Proc. of ANTS-I, Lecture Notes in Computer Science 877, pp.43-58, 1994.
6. J. M. Couveignes, L. Dewaghe and F. Morain, "Isogeny cycles and the Schoof-Elkies-Atkin algorithm," LIX/RR/96/03, 1996.
7. T. Denny, O. Schirokauer and D. Weber, "Discrete logarithms : the effectiveness of the index calculus method," in the Proc. of ANTS-II, Lecture Notes in Computer Science 1122, pp.337-362, Springer-Verlag, 1996.

8. N. J. Elkies, "Explicit isogenies," Draft, 1991.
9. G. Frey and H. G. Rück, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," Math. Comp., 62, 206, pp.865-874, 1994.
10. G. Frey, M. Müller and H. G. Rück, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," preprint, 1998.
11. R. Harasawa, J. Shikata, J. Suzuki and H. Imai, "Comparing the MOV and FR reductions in elliptic curve cryptography," Advances in Cryptology - EUROCRYPT'99, Lecture Notes in Computer Science, vol.1592, pp.190-205, 1999.
12. N. Koblitz, "Elliptic Curve Cryptosystems," Math. Comp. 48, pp.203-209, 1987.
13. N. Koblitz, "Algebraic Aspects of Cryptography," Springer-Verlag, 1998.
14. R. Lercier and F. Morain, "Counting the number of points on elliptic curves over finite fields : strategy and performances," Advances in Cryptology - EUROCRYPT'95, Lecture Notes in Computer Science, vol.921, pp.79-94, 1995.
15. R. Lercier, "Computing isogenies in F_{2^n}," In the Proc. of ANTS-II, Lecture Notes in Computer Science 1122, Springer, pp.197-212, 1996.
16. A. Menezes, "Elliptic Curve Public Key Cryptosystem," Kluwer Acad. Publ., Boston, 1993.
17. A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms in a finite field," IEEE Transactions on Information Theory, vol.IT-39, no.5, pp.1639-1646, 1993.
18. V. Miller, "Use of elliptic curves in cryptography," Advances in Cryptology - CRYPTO'85, Lecture Notes in Computer Science, vol.218, pp.417-426, Springer, 1986.
19. V. Miller, "Short programs for functions on curves," unpublished manuscript, 1986.
20. F. Morain, "Calcul du nombre de points sur une courbe elliptique dans un corps fini : aspects algorithmiques," J. Théor. Nombres Bordeaux 7, pp.255- 282, 1995..
21. T. Satoh and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves," Commentarii Math. Univ. St. Pauli, 47(1), pp.81-92, 1998.
22. R. Schoof, "Nonsingular plane cubic curves over finite fields," J. Combinatorial Theory, Series A, vol.46, pp.183-211, 1987.
23. R. Schoof, "Counting points on elliptic curves over finite fields," J. Théor. Nombres Bordeaux 7, pp.219-254, 1995.
24. I. Semaev, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p," Math. of Computation 67, pp.353-356, 1998.
25. J. Silverman, "The Arithmetic of Elliptic Curves," Springer-Verlag, New York, 1986.
26. J. Silverman and J. Suzuki, "Elliptic curve discrete logarithms and index calculus," Advances in Cryptology - ASIACRYPT'98, Lecture Notes in Computer Science 1514, Springer, pp.110-125, 1998.
27. N. Smart, "The Discrete logarithm problem on elliptic curves of trace one," to appear in Journal of Cryptology.

Appendix A (Proof of Lemma 1)

Proof of Lemma 1. The proof is completed by induction on i.

The case $i = 0$ is trivial. We consider the case $i = 1$. Set $S' := f^{(1)}(S) = (\phi - 1)S$ and $T' := f^{(1)}(T) = (\phi - 1)T$. We first show that the orders of S' and T' are n^{r-1} and n^s, respectively. Clearly, $n^{r-1}S' = (\phi - 1)(n^{r-1}S) = O$. Since

$$n^j S' = O \Leftrightarrow (\phi - 1)(n^j S) = O \Leftrightarrow n^j S \in \langle P \rangle = \langle n^{r-1}S \rangle,$$

it follows that $j \geq r - 1$. Thus, the order of S' is n^{r-1}. Also, clearly, $n^s T' = (\phi - 1)(n^s T) = O$. To prove that the order of T' is n^s, it is sufficient to show that $n^j T' \neq O$ for any $j < s$. Suppose on the contrary that $n^j T' = O$, then we have $n^j T \in \langle P \rangle = \langle n^{r-1}S \rangle$. This contradicts the assumption that S and T are algebraically independent. We next show that S' and T' are algebraically independent. Suppose on the contrary that there is a non-trivial relation

$$n^m(an^{r-1-s}S' + bT') = O, \quad (\mathrm{GCD}(a, n) = 1,\ \mathrm{GCD}(b, n) = 1,\ 0 \leq m < s).$$

(Note that any non-trivial relation can be expressed as above since the orders of S' and T' are n^{r-1} and n^s, respectively.) Then we have

$$\begin{aligned} n^m\{an^{r-1-s}(\phi - 1)(S) + b(\phi - 1)(T)\} = O &\Leftrightarrow (\phi - 1)(n^m(an^{r-1-s}S + bT)) = O \\ &\Leftrightarrow n^m(an^{r-1-s}S + bT) \in \langle P \rangle = \langle n^{r-1}S \rangle. \end{aligned}$$

Therefore, there is some $c \in Z/nZ$ such that $n^m(an^{r-1-s}S + bT) = cn^{r-1}S$. The multiplication by n^{s-m} on the both sides of the above equation induces $an^{r-1}S = O$, which is a contradiction since the order of S is n^r and $\mathrm{GCD}(a, n) = 1$.

Assume that the statement of the lemma is true for $i - 2$ and $i - 1$. We first show that the orders of $f^{(i)}(S)$ and $f^{(i)}(T)$ are n^{r-i} and n^s, respectively. From the induction hypothesis that $f^{(i-1)}(S)$ has order n^{r-i+1} and that $r - i + 1 > s$, we can represent it in the form

$$f^{(i-1)}(S) = an^{i-1}S + bT, \quad (\mathrm{GCD}(a, n) = 1,\ b \in Z/n^s Z).$$

Therefore, we have

$$\begin{aligned} n^{r-i}f^{(i)}(S) &= n^{r-i}(\phi - 1)f^{(i-1)}(S) \\ &= n^{r-i}(\phi - 1)(an^{i-1}S + bT) \\ &= (\phi - 1)(an^{r-1}S + bn^{r-i}T) \\ &= (\phi - 1)(an^{r-1}S) = O. \end{aligned}$$

(Note that $n^{r-i}T = O$ since we now consider the case $i \leq r - s \Leftrightarrow s \leq r - i$.) Also, if $n^j f^{(i)}(S) = O$, then

$$(\phi - 1)(n^j f^{(i-1)}(S)) = O \Leftrightarrow n^j f^{(i-1)}(S) \in \langle P \rangle = \langle n^{r-1}S \rangle.$$

It follows that $j + 1 \geq r - i + 1 \Leftrightarrow j \geq r - i$ since $n^{j+1}f^{(i-1)}(S) = O$. Thus, the order of $f^{(i)}(S)$ is n^{r-i}. On the other hand, from the induction hypothesis that the order of $f^{(i-1)}(T)$ is n^s, it follows that $n^s f^{(i)}(T) = (\phi - 1)(n^s f^{(i-1)}(T)) = O$.

To prove that the order of $f^{(i)}(T)$ is n^s, it is sufficient to show that $n^j f^{(i)}(T) \neq O$ for any $j < s$. Suppose on the contrary that $n^j f^{(i)}(T) = O$ for some $j < s$, then we have

$$n^j(\phi - 1)f^{(i-1)}(T) = O \Leftrightarrow n^j f^{(i-1)}(T) \in \langle P \rangle = \langle n^{r-1}S \rangle.$$

Therefore, $n^{j+1}f^{(i-1)}(T) = O$, from which it follows that $j + 1 \geq s$. Thus, we obtain $j = s - 1$ (note that $j < s$), and furthermore, $n^{s-1}f^{(i-1)}(T) = an^{r-1}S$ for some $a \in Z/nZ)^{\times}$. Then we have

$$n^{s-1}f^{(i-1)}(T) = an^{r-1}S \Leftrightarrow (\phi - 1)(n^{s-1}f^{(i-2)}(T)) = an^{r-1}S,$$

and it follows that

$$\phi(n^{s-1}f^{(i-2)}(T)) = n^{s-1}f^{(i-2)}(T) + an^{r-1}S. \tag{2}$$

Thus we have

$$\begin{aligned} O &= n^{s-1}f^{(i)}(T) \\ &= n^{s-1}(\phi - 1)^2(f^{(i-2)}(T)) \\ &= n^{s-1}(\phi^2 - 2\phi + 1)(f^{(i-2)}(T)) \\ &= n^{s-1}\{(t-2)\phi(f^{(i-2)}(T)) + (1-q)f^{(i-2)}(T)\} \qquad (\text{since } \phi^2 = t\phi - q) \\ &= (t-2)\{n^{s-1}f^{(i-2)}(T) + an^{r-1}S\} + (1-q)(n^{s-1}f^{(i-2)}(T)) \quad (\text{since } (2)) \\ &= (t-1-q)(n^{s-1}f^{(i-2)}(T)) + (t-2)an^{r-1}S \\ &= (t-2)an^{r-1}S, \end{aligned}$$

where the last equality follows from the assumption that $n | \#E(F_q) = 1 + q - t$ and that the order of $f^{(i-2)}(T)$ is n^s. Thus, we obtain $t - 2 \equiv 0 \bmod n$ since $a \in (Z/nZ)^{\times}$. Therefore, it follows that $q - 1 \equiv 0 \bmod n$. This contradicts the assumption that $n \not| q - 1$.

We next show that $f^{(i)}(S)$ and $f^{(i)}(T)$ are algebraically independent. (This is proved similarly as in the case $i = 1$.) Suppose on the contrary that there is a non-trivial relation

$$n^m\{an^{r-i-s}f^{(i)}(S) + bf^{(i)}(T)\} = O \quad (\mathrm{GCD}(a, n) = 1,\ \mathrm{GCD}(b, n) = 1,\ 0 \leq m < s).$$

(Note that the orders of $f^{(i)}(S)$ and $f^{(i)}(T)$ are n^{r-i} and n^s, respectively, and that $r - i \geq s$.) Then we have

$$\begin{aligned} & n^m(\phi - 1)(an^{r-i-s}f^{(i-1)}(S) + bf^{(i-1)}(T)) = O \\ \Leftrightarrow\ & n^m(an^{r-i-s}f^{(i-1)}(S) + bf^{(i-1)}(T)) \in \langle P \rangle = \langle n^{r-1}S \rangle. \end{aligned}$$

Therefore, the multiplication by n^{s-m} on the above last formula induces

$$\begin{aligned} & n^s\{an^{r-i-s}f^{(i-1)}(S) + bf^{(i-1)}(T)\} = O \\ \Leftrightarrow\ & an^{r-i}f^{(i-1)}(S) = O, \end{aligned}$$

which is a contradiction since $f^{(i-1)}(S)$ has order n^{r-i+1} and $\mathrm{GCD}(a, n) = 1$. The proof is completed. □