

Sublinear Scalar Multiplication on Hyperelliptic Koblitz Curves

Hugo Labrande¹ and Michael J. Jacobson Jr.^{2,*}

¹ ENS Lyon

46 Allée d'Italie, 69364 Lyon Cedex 07, France

`hugo.labrande@ens-lyon.fr`

² Department of Computer Science, University of Calgary

2500 University Drive NW, Calgary, Alberta, Canada T2N 1N4

`jacobs@cpsc.ucalgary.ca`

Abstract. Recently, Dimitrov et. al. [5] proposed a novel algorithm for scalar multiplication of points on elliptic Koblitz curves that requires a provably sublinear number of point additions in the size of the scalar. Following some ideas used by this article, most notably double-base expansions for integers, we generalize their methods to hyperelliptic Koblitz curves of arbitrary genus over any finite field, obtaining a scalar multiplication algorithm requiring a sublinear number of divisor additions.

Keywords: Hyperelliptic Koblitz curves, scalar multiplication, double-base expansions.

1 Introduction

Following an idea proposed independently by Koblitz and Miller in 1985 (respectively in [9] and [12]), elliptic and hyperelliptic curves over finite fields are now widely used for cryptographic purposes. Operations are carried out in the finite group of rational points of the elliptic curve; schemes such as ElGamal encryption can be applied in this group. Due to particular properties of the group, cryptography on elliptic curves offers security as strong as other algorithms (such as the RSA algorithm, for instance) with keys of shorter lengths. Hyperelliptic curve cryptosystems are interesting generalizations of elliptic curve cryptosystems. Curves with a too high genus were shown to be cryptographically insecure [7], but if the genus is small hyperelliptic curves can be used in some cryptosystems, and represent an alternative that is as secure and efficient as elliptic curve cryptosystems.

Anomalous binary elliptic curves (elliptic Koblitz curves), first introduced by Koblitz in a 1992 article [10], are of particular interest in cryptographic applications. Cryptographic schemes using elliptic curves are faster on Koblitz curves than on any other type of curves. Moreover, secure Koblitz curves are easy to find: all this make those curves very convenient for cryptography. Those

* Supported in part by NSERC of Canada.

curves can be used for instance in embedded systems, where the computing power and the memory are limited.

The problem we deal with in this article is scalar multiplication, computing m -folds of elements of the group associated to the curve. This problem has practical implications: cryptographic schemes in the group of an elliptic or hyperelliptic curve often require computing m -folds. Thus, by making scalar multiplication more efficient, we improve the speed of curve-based cryptosystems, possibly making them more practical and applicable in a broader set of systems.

In [5] the authors present a method to compute m -folds of points ($m \in \mathbb{Z}$) on elliptic Koblitz curves requiring a sublinear (in the size of m) number of point additions. The method involves finding a suitable triple-base τ -adic expansion for a particular complex number τ . They also design an algorithm using double-base expansions that, while not requiring a provably sublinear number of point additions (although benchmarks and experimental results would let us think it is), is efficient in practice. These represent the first result that breaks the barrier of linearity in number of point additions; all previous algorithms were using a linear number of them, and aimed at improving the O -constant.

In this paper, we generalize the methods of [5] to the case of scalar multiplication on hyperelliptic Koblitz curves of all characteristic and all genera. We present a scalar multiplication algorithm with sublinear complexity (in the number of divisor additions) using triple-base expansions, with carefully chosen bases. Although mostly of theoretical interest due to large constants in the O -notation, our algorithm does prove for the first time the existence of sublinear divisor multiplication on hyperelliptic Koblitz curves. We also present an algorithm using double-base expansions that is conjecturally sublinear and more likely to perform well in practice.

The next two sections provide background on hyperelliptic Koblitz curves and multi-base number systems. The sections that follow contain our results. We first present our algorithm that uses triple-base expansions to achieve a sublinear complexity in number of divisor additions, followed by our practical double-base algorithm.

2 Hyperelliptic Koblitz Curves

More information on hyperelliptic curves can be found, for example, in [2, pp.81-85].

Let $q = p^r$ be a prime power and let \mathbb{F}_q be the finite field with q elements. A (non-singular) hyperelliptic curve of genus g with one point at infinity is defined by the equation

$$C : v^2 + h(u)v = f(u),$$

where $h, f \in \mathbb{F}_q[X]$, $\deg(h) \leq g$, f monic of degree $2g+1$, and if $y^2 + h(x)y = f(x)$ for $(x, y) \in \overline{\mathbb{F}_q} \times \overline{\mathbb{F}_q}$, then $2y + h(x) \neq 0$ or $h'(x)y - f'(x) \neq 0$.

Let $\text{Pic}^0(C(\mathbb{F}_q))$ denote the degree zero divisor class group of C over \mathbb{F}_q . Elements of $\text{Pic}^0(C(\mathbb{F}_q))$ can be represented uniquely using the Mumford representation, a pair of polynomials $[u, v]$, $u, v \in \mathbb{F}_q[x]$, $\deg v < \deg u \leq g$, u monic,

and $u|f - v^2 - hv$. Thus, each divisor can be represented by at most $2g$ elements of \mathbb{F}_q . The divisor corresponding to the principal ideals class is denoted $\text{div}[1, 0]$. The inverse of $[u, v]$ is $[u, -h - v]$, where the second entry is reduced modulo u , and is thus efficiently computable. The group operation can be done using Cantor’s algorithm for any genus; for genus up to 4 more efficient explicit formulas exist.

Hyperelliptic Koblitz curves are hyperelliptic curves defined over \mathbb{F}_q but the group $\text{Pic}^0(C(\mathbb{F}_{q^n}))$ is considered over \mathbb{F}_{q^n} where n is prime. For example, one of the hyperelliptic curves that is studied in [8] is the curve of genus 2

$$C : v^2 + uv = u^5 + u^2 + 1$$

considered over \mathbb{F}_{2^n} . Such curves are a generalization of the approach developed in the elliptic case by Koblitz [10]. In a string of articles, some authors successfully generalized Solinas’s scalar multiplication method [13] to hyperelliptic curves to get fast algorithms for divisor multiplication on this type of Koblitz curve. An article [8] describes the method for hyperelliptic curves of genus 2, and subsequent work by Lange describes a generalization of the method for hyperelliptic curves of all genera and for every characteristic [11] (see also [2, Sections 15.1.2 and 15.1.3]).

As in the elliptic case, one main interest in hyperelliptic Koblitz curves is that scalar multiplication can be sped up by making use of the action of the Frobenius endomorphism on elements in $\text{Pic}^0(C(\mathbb{F}_q))$. The Frobenius endomorphism τ over \mathbb{F}_{q^n} is defined as $x \rightarrow x^q$. This operation is inherited by points on the curve and by $\text{Pic}^0(C(\mathbb{F}_q))$. It operates on elements of $\text{Pic}^0(C(\mathbb{F}_q))$ given in Mumford representation by

$$\tau([u(x), v(x)]) = [\tau(u(x)), \tau(v(x))], \quad \text{where } \tau\left(\sum_{i=0}^d u_i x^i\right) = \sum_{i=0}^d u_i^q x^i .$$

In this manner, τ acts as an endomorphism on the group $\text{Pic}^0(C(\mathbb{F}_q))$.

To generalize the methods discussed previously, we have to represent the Frobenius endomorphism as a complex number τ . Let P be the characteristic polynomial of the Frobenius endomorphism:

$$P(T) = T^{2g} + a_1 T^{2g-1} + \dots + a_g T^g + a_{g-1} q T^{g-1} + \dots + a_1 q^{g-1} T + q^g .$$

Let τ be a complex root of P ; since the Frobenius endomorphism and τ are both roots of this polynomial, we can consider the Frobenius endomorphism as the complex number τ . For example, in the case of the genus 2 Koblitz curve

$$C_1 : v^2 + uv = u^5 + u^2 + 1 ,$$

we may take $\tau = \frac{\mu \pm i\sqrt{4-\mu}}{2}$, where $\mu = \frac{1 \pm \sqrt{17}}{2}$.

The idea to improve scalar multiplication is to compute a base- τ expansion of the scalar, enabling a version of binary exponentiation based on repeated

applications of τ as opposed to doublings. As the computation of τ is negligible compared to the cost of a doubling, this yields an especially efficient algorithm.

The problem of finding a τ -adic representation using this set of coefficients is addressed in [11, Algorithm 5.19]. The algorithm attempts to compute a τ -adic expansion of a given scalar using the digit set $R = \{0, \pm 1, \dots, \pm \lceil \frac{q^g-1}{2} \rceil\}$. Lange proved [11, Theorem 5.5] that, unlike the elliptic case, the expansions produced by this algorithm are not necessarily finite. Some criteria for non-finiteness are provided in particular cases, and in general it is possible to check for a particular curve whether periodic expansions occur by testing a set of elements in $\mathbb{Z}[\tau]$ of bounded norm. These results use the following norm, which we also use in this paper:

$$\mathcal{N}(\zeta) = \sqrt{\sum_{i=1}^g \left| \sum_{j=0}^{2g-1} b_j \tau_i^j \right|^2},$$

where $\zeta = b_0 + b_1\tau + \dots + b_{2g-1}\tau^{2g-1} \in \mathbb{Z}[\tau]$ and $\tau_1, \tau_2, \dots, \tau_g$ denote g conjugates of τ (one of each conjugate pair).

In the case that expansions are finite, Lange proves [11, Theorem 5.16] that the number of terms is bounded by $n + 4g + 1$. This bound is achieved by first reducing the scalar modulo $\frac{\tau^n-1}{\tau-1}$ in $\mathbb{Z}[\tau]$, as in the elliptic case. The expected number of non-zero terms in the expansion is $\frac{q^g-1}{q^g}$, yielding an algorithm that requires the precomputation of $\lceil \frac{q^g-1}{2} \rceil$ divisors and only $\frac{q^g-1}{q^g}(n + 4g + 1)$ divisor additions on average.

When compared to standard methods, Lange’s algorithm leads to a speed-up of $1.5g$ as compared to the binary method and $1.3g$ compared to NAF. When compared to a binary window method of width 2 (assuming $q = 2$ and $g = 2$), the speed-up is $11/3$. However, we notice that the asymptotic complexity (in number of divisor additions) is still linear in the size of the scalar m assuming, as is usually the case in cryptographic applications, that $m \in O(q^{ng})$. In the next sections, we give two algorithms that achieve sublinear complexity, one provably so and the other conjecturally.

3 Multi-base Number Systems

As in [5], the main tool we use to achieve sublinearity is multi-base expansions of elements of $\mathbb{Z}[\tau]$.

Definition 1 (double-base expansions). *Let $P, Q, m \in \mathbb{Z}[\tau]$. An expression of the form:*

$$m = \sum_{i=1}^d r_i P^{a_i} Q^{b_i},$$

where $0 \neq r_i \in R \subset \mathbb{N}$ and $a_i, b_i \in \mathbb{Z}_{\geq 0}$, is called a double-base representation or $\{P, Q\}$ -representation of m .

Definition 2 (triple-base expansion). *Let $P, Q, S, m \in \mathbb{Z}[\tau]$. An expression of the form:*

$$m = \sum_{i=1}^d r_i P^{a_i} Q^{b_i} S^{c_i} \ ,$$

where $0 \neq r_i \in R \subset \mathbb{N}$ and $a_i, b_i, c_i \in \mathbb{Z}_{\geq 0}$, is called a triple-base representation or $\{P, Q, S\}$ -representation of m .

Our definitions are adapted from [3,5], where integer scalars are considered, and the digit set $R = \{\pm 1\}$.

The motivation of applying multi-base expansions to the scalar multiplication problem is that the number of non-zero terms in such an expansion, when using appropriate bases, is sublinear in the size of the scalar. In the case of integer bases and scalars, we have the following theorem.

Theorem 1. *Given two primes p, q , every integer m has a $\{p, q\}$ -representation with a sublinear number of summands, i.e., it can be represented as the sum or difference of at most $O(\frac{\log m}{\log \log m})$ integers of the form $p^a q^b$ for $a, b \in \mathbb{N}$ with $a, b \in O(\log m)$.*

Theorem 1 first appeared with proof in [3, Theorem 1] for bases $p = 2$ and $q = 3$, but generalizes to any number of arbitrary distinct prime bases. The representation can be computed using a greedy algorithm, namely computing the largest integer of the form $p^a q^b$ less than or equal to m and repeating with $m - p^a q^b$. A result of Tijdeman [14] implies that there exists $p^a q^b$ with $m - m/(\log m)^C < p^a q^b < m$ for some absolute constant $C > 0$; this implies the running time and the bound on the exponents a and b occurring in the representation of m .

Tijdeman’s result also holds for complex bases provided that the norm of one base is strictly greater than the other. For elliptic and hyperelliptic Koblitz curves, we would like to use bases that are simple functions of τ , ideally τ and $\tau - 1$, so that the resulting scalar multiplication algorithm requires as few divisor additions as possible. Unfortunately these bases have the same norm, and the theoretical result does not apply. In [5], the authors get around this problem by using a combination of triple-base representations in $\mathbb{Z}[\tau]$ and $\{2, 3\}$ -representations of integers, yielding an algorithm requiring only $o(\log m)$ point additions. As that algorithm does not appear to be efficient in practice, an algorithm using $\{\tau, \tau - 1\}$ -representations is also presented that works well in practice, despite having only conjectural sublinearity.

4 A Sublinear Divisor Multiplication Algorithm Using Triple-Base Expansions

Our goal is to find a representation of an integer with a sublinear number of summands that leads to a sublinear scalar multiplication algorithm. By sublinear, we mean that the number of divisor additions is sublinear in the size of the

integer. Our asymptotic statements in this section assume that the field size q and genus g are fixed, so that the norm of the scalar tends to infinity, although we also give the dominant terms in q and g as well.

In [5], the authors achieve this for elliptic curves by using $\{2, 3\}$ -expansions of integers and then replacing the 2s and 3s by expressions involving τ using

$$\begin{aligned} 2 &= \tau(\mu - \tau) \\ 3 &= 1 + \mu\tau - \tau^2 \end{aligned} ,$$

where $\mu \in \{\pm 1\}$ is a parameter indicating which of the two elliptic Koblitz curves over \mathbb{F}_2 is used. We use a similar approach. We first select suitable bases for our representation, that is two prime numbers that we can replace by polynomial identities involving τ . Considering the characteristic polynomial of the Frobenius endomorphism τ , we have the following identity:

$$q^g = -\tau^{2g} - a_1\tau^{2g-1} - \dots - a_1q^{g-1}\tau = Q(\tau).$$

Consider $i, j \in \mathbb{Z}$ such that $q^g + i$ and $q^g + j$ are prime. By Theorem 1, any integer can be represented with a sublinear number of summands of the form $\pm(Q(\tau) + i)^x(Q(\tau) + j)^y$. For convenience we call $\{\tau, Q(\tau) + i, Q(\tau) + j\}$ -integers terms of the form $\pm\tau^x(Q(\tau) + i)^y(Q(\tau) + j)^z$, $x, y, z \in \mathbb{Z}$.

First, note that the straightforward approach of computing a $\{q^g + i, q^g + j\}$ representation of the integer scalar m and performing the substitution does not yield a sublinear algorithm. Although the number of terms in the expansion is indeed sublinear, the number of required divisor additions may not be because the required powers of $q^g + i$ and $q^g + j$ may be as large as $\log m$. Instead, we model our approach on that of [5] and obtain the following theorem.

Theorem 2. *Let $\zeta \in \mathbb{Z}[\tau]$, and assume that the τ -adic representation of ζ with coefficients in $R = \{0, \pm 1, \dots, \pm \lfloor \frac{q^g - 1}{2} \rfloor\}$ is finite. Then, for g and q fixed, ζ can be represented as the sum of at most*

$$O\left(gq^g \frac{\log \mathcal{N}(\zeta)}{\log \log \mathcal{N}(\zeta)}\right)$$

$\{\tau, Q(\tau) + i, Q(\tau) + j\}$ -integers such that the largest power of both $Q(\tau) + i$ and $Q(\tau) + j$ is $O(q^g [g + \log \mathcal{N}(\zeta)]^\alpha)$ for any real constant α such that $0 < \alpha < 1/2$.

Proof. Let $\alpha \in (0, 1/2)$. We first determine the τ -adic representation of ζ using coefficients taken in R by using Algorithm 5.19 of [11]. As we assume the length of this expansion is finite, Lemma 5.6 of [11] implies (see the discussion on p.58 of [11]) that its length is $l = O(\log \mathcal{N}(\zeta) + g)$. For convenience, we denote $N_0 = g + \log_q \mathcal{N}(\zeta)$.

Now, we break this representation in $M = \lceil N_0^{1-\alpha} \rceil$ blocks of $O(N_0^\alpha)$ coefficients each such that

$$\zeta = \sum_{i=1}^l x_i \tau^i = \sum_{i=0}^{M-1} C_i \tau^{ik} \text{ (where } k = \lfloor l/M \rfloor \text{)}.$$

Using the fact that $P(\tau) = 0$, we see that for $i \in \{0 \dots M\}$, the i th block corresponds to an element of the form

$$C_i = \sum_{j=0}^{2g-1} c_{ij} \tau^i.$$

We note that, since the x_i are in R (and thus bounded by $O(q^g)$) and that there are $O(N_0^\alpha)$ digits in each block, $\log c_{ij}$ is in $O(q^g N_0^\alpha)$.

We represent each c_{ij} in double-base representation using the prime integer bases $q^g + i$ and $q^g + j$. According to Theorem 1, and since both of our bases are prime, these representations can be computed using the greedy algorithm of [3], and have at most

$$O\left(\frac{q^g N_0^\alpha}{\log q^g N_0^\alpha}\right)$$

summands of the form $(q^g + i)^x (q^g + j)^y$ where $x, y \in O(q^g N_0^\alpha)$. Then, since $q^g = Q(\tau)$, we substitute $q^g + i$ by $Q(\tau) + i$ and $q^g + j$ by $Q(\tau) + j$ to obtain a $\{Q(\tau) + i, Q(\tau) + j\}$ -expansion of each c_{ij} .

Next, we compute the $\{\tau, Q(\tau) + i, Q(\tau) + j\}$ -expansions of C_i by multiplying the expansion of c_{ij} for each $j \in \{0 \dots 2g - 1\}$ by τ^j , and adding the results. Thus, the expansion of C_i has $O(gq^g N_0^\alpha / \log q^g N_0^\alpha)$ summands of the form $\pm \tau^x (Q(\tau) + i)^y (Q(\tau) + j)^z$, with $x \in \{0 \dots 2g - 1\}$ and $y, z \in O(q^g N_0^\alpha)$.

The last step is to compute the expansion of ζ from the expansions of the M blocks C_i , by multiplying each C_i by τ^{il} . We obtain a $\{\tau, Q(\tau) + i, Q(\tau) + j\}$ -expansion of ζ that has

$$O\left(gq^g \frac{N_0^\alpha}{\log q^g N_0^\alpha} N_0^{1-\alpha}\right) = O\left(gq^g \frac{N_0}{\log q^g N_0^\alpha}\right)$$

terms, and in which the exponents of $Q(\tau) + i$ and $Q(\tau) + j$ are $O(q^g N_0^\alpha)$. Now, since

$$\begin{aligned} \log q^g N_0^\alpha &= g \log q + \log(g + \log \mathcal{N}(\zeta))^\alpha \\ &\geq \alpha \log(g + \log \mathcal{N}(\zeta)) \\ &\geq \alpha \log \log \mathcal{N}(\zeta) \end{aligned}$$

assuming that g and q are fixed, we get that our number of terms in the end is indeed

$$O\left(gq^g \frac{g + \log \mathcal{N}(\zeta)}{\alpha \log \log \mathcal{N}(\zeta)}\right) = O\left(gq^g \frac{\log \mathcal{N}(\zeta)}{\log \log \mathcal{N}(\zeta)}\right)$$

as required. □

The proof of Theorem 2 is constructive in the sense that it leads immediately to an algorithm to compute a $\{\tau, Q(\tau) + i, Q(\tau) + j\}$ -expansion of $\zeta \in \mathbb{Z}[\tau]$.

This leads to the following algorithm for computing ζD using a sublinear number of divisor additions in $\log \mathcal{N}(\zeta)$. The idea is, given the representation of

ζ from Theorem 2, to compute $(Q(\tau) + i)^a(Q(\tau) + j)^bD$ for all powers a, b in the representation, use these to compute each term in the representation multiplied by D , and then to add these together. We will prove that the sublinearity holds for fixed g and q as long α is selected satisfying $0 < \alpha < 1/2$.

Algorithm 1. Divisor multiplication using triple-base expansions

Input: $\zeta = b_0 + b_1\tau + \dots + b_{2g-1}\tau^{2g-1}$ and $D \in \text{Pic}^0(C(\mathbb{F}_q))$

Output: ζD

- 1: Find $i, j \in \mathbb{Z}$ such that $q^g + i$ and $q^g + j$ are prime.
 - 2: Compute a $\{\tau, Q(\tau) + i, Q(\tau) + j\}$ -expansion of ζ using Theorem 2 with any α such that $0 < \alpha < 1/2$; store terms in $L = \{s_i\tau^{a_i}(Q(\tau) + i)^{b_i}(Q(\tau) + j)^{c_i} \text{ with } 1 \leq i \leq d, s_i = \pm 1, a_i, b_i \in \mathbb{Z}_{\geq 0}\}$.
 - 3: Compute $A = \max(b_i)$ and $B = \max(c_i)$.
 - 4: Compute in succession for $x \in \{0 \dots A\}$ the divisor classes $D_x = (Q(\tau) + i)D_{x-1}$, with $D_0 = D$.
 - 5: **for** $x \in \{0 \dots A\}$ **do**
 - 6: Compute in succession for $y \in \{0 \dots B\}$ the divisor classes $F_{x,y} = (Q(\tau) + j)F_{x,y-1}$, with $F(x, 0) = D_x$.
 - 7: **end for**
 - 8: $Res \leftarrow \text{div}[1, 0]$
 - 9: **for** $i = 1, \dots, d$ **do**
 - 10: $Res \leftarrow Res + s_i\tau^{a_i}F_{b_i, c_i}$
 - 11: **end for**
 - 12: **return** Res
-

Theorem 3. Algorithm 1 requires $o(gq^g \log \mathcal{N}(\zeta))$ divisor additions for fixed g and q , i.e. the required number of divisor additions is sublinear in $\log \mathcal{N}(\zeta)$.

Proof. We analyze the algorithm step by step:

1. Step 1: this step does not require any divisor additions. We give here an order of magnitude of i and j . By Chebyshev’s theorem, there is a prime number between n and $2n$ for any integer n . Thus, we know there is a prime number between $\lfloor q^g/2 \rfloor$ and q^g , and between q^g and $2q^g$, and we can bound $|i|$ and $|j|$ by q^g .
2. Steps 2-3: these steps also require no divisor additions. Note that the greedy algorithm of [3] can be used to compute the double-base representations of the c_{ij} , and that consequently $A, B \in O(\log^\alpha \mathcal{N}(\zeta))$.
3. Step 4: we compute A divisors, each one being derived from the previous one by applying $Q(\tau) + i$ to it. Applying $Q(\tau) + i$ to a divisor D can be done as follows:
 - (a) Compute rD for every $r \leq 2q^g$. Since the absolute value of every coefficient in $Q(\tau)$ is smaller than $2^{2g}q^{g/2}$ (see [15, p.378]) and $|i| \leq q^g$, every coefficient of $Q(\tau) + i$ is bounded by $2q^g$. This step requires $O(q^g)$ divisor additions.

- (b) Compute every term in $Q(\tau) + i$ by application of the Frobenius endomorphism and add those terms together. This step requires $O(g)$ divisor additions.

Thus, the complexity of this step is $O(q^g \log^\alpha \mathcal{N}(\zeta))$ divisor additions.

4. Step 5 to 7: we compute AB divisors by repeatedly applying $Q(\tau) + j$ to the divisors computed in the previous step. As discussed in the analysis of previous step (since we have $|j| \leq q^g$ as well), each of those require $O(q^g)$ divisor additions. The complexity of this step is $O(q^g \log^{2\alpha} \mathcal{N}(\zeta))$ divisor additions.
5. Step 8 to 11: Since $s = \pm 1$, the number of point additions is equal to the number of terms of the expansion, which by Theorem 2 is

$$O\left(gq^g \frac{\log \mathcal{N}(\zeta)}{\log \log \mathcal{N}(\zeta)}\right).$$

Now, since $\alpha < 1/2$, the total number of divisor additions in Steps 4 and 5 is $o(q^g \log \mathcal{N}(\zeta))$. Thus, the number of divisor additions for the entire algorithm is $o(q^g \log \mathcal{N}(\zeta))$. □

Note that, although the number of divisor additions required is sublinear, the overall bit complexity of the algorithm is linear in $q^g \log \mathcal{N}(\zeta)$. This is due to the cost of computing the representation in Step 2. The first step of Theorem 2 is to compute the τ -adic expansion of ζ , which has complexity $O(\log \mathcal{N}(\zeta))$. In addition, double base expansions of the c_{ij} must be computed. From [4], the bit complexity of each of these operations is in $O(\log c_{ij}) = O(q^g N_0^\alpha)$. There are $2g \times M = 2g \lceil N_0^{1-\alpha} \rceil$ of the c_{ij} , so the total bit complexity is in $O(gq^g N_0) = O(gq^g \log \mathcal{N}(\zeta))$.

A straightforward application of this algorithm to $\zeta = m \in \mathbb{Z}$ allows one to compute mP in $o(gq^g \log m)$ divisor additions, as $\log \mathcal{N}(m) = \log \sqrt{gm^2} \in O(\log m)$. However, in the case that m is of the usual size used for cryptographic applications, namely $O((q^n)^g)$, we can do better by first reducing it modulo $\tau^n - 1$, as $\tau^n(D) = D$ in $\text{Pic}^0(C(\mathbb{F}_{q^n}))$. If, as is also usual in cryptographic applications, arithmetic is restricted to a large prime order subgroup of $\text{Pic}^0(C(\mathbb{F}_{q^n}))$, we can reduce the scalar by $\frac{\tau^n - 1}{\tau - 1}$ (see [11, p.65]). For $M \equiv m \pmod{\frac{\tau^n - 1}{\tau - 1}}$, we get $\log \mathcal{N}(M) = O(n + 2g) = O\left(\frac{\log m}{g \log q} + 2g\right)$, as we are assuming that $\log m = \log q^{ng}$. Thus, by applying our algorithm to M instead of m we require the same number of divisor additions asymptotically, but would save a factor of g .

Although our algorithm is sublinear in $\log m$, it depends badly on q and g . However, the most typical applications of Koblitz curves for cryptographic purposes are with small q (to enable easy computation of group orders) and small g (because $g > 3$ is insecure — see, for example, [2, Section 23.2.1]). Note also that our result coincides with that of [5] for the elliptic case, where $q = 2$ and $g = 1$.

We note that this algorithm is asymptotically more efficient than previous methods such as the double-and-add method ($3/2gn \log q$ divisor additions) or

Lange's method using single-base expansion ($(1-1/q^g)n$ divisor additions), since its complexity is sublinear ($o(g^2q^g n)$). However, the presence of big constants in the asymptotic complexity makes it likely less efficient than those algorithms in practice; we stress that the point of this algorithm was to prove that there exists a scalar multiplication algorithm that only requires a sublinear number of divisor additions. Furthermore, we note that our asymptotics in the case $g = 1$ (elliptic case) and $q = 2$ is the same as in [5] – the primes being used in both methods being 2 and 3.

5 A Practical Scalar Multiplication Algorithm Using Double-Base Expansion

In [5], the authors also devise a scalar multiplication algorithm for elliptic Koblitz curves using $\{\tau, \tau - 1\}$ -expansions that is designed to work well in practice. Even though these bases have the same norm, and thus cannot be proved to yield sublinear length representations using the results of [14], they were selected because they are as cheap as possible to apply to a given point (0 or 1 addition required). A greedy algorithm to compute representations is too expensive, as it is not known how to efficiently compute the closest $\{\tau, \tau - 1\}$ number to a given element in $\mathbb{Z}[\tau]$. Hence, a blocking strategy is used, in which each short block of a τ -adic representation is replaced by a pre-computed optimal $\{\tau, \tau - 1\}$ representation. Assuming that these bases do yield sublinear representations, it is proved that the strategy yields a sublinear algorithm, and numerical results were presented demonstrating its efficiency in practice.

We attempt to follow the same strategy with hyperelliptic Koblitz curves, using bases τ and $\tau - 1$, and terms of the form $r_i \tau^{a_i} (\tau - 1)^{b_i}$, with $r \in R = \{0, \pm 1, \dots, \pm \lceil \frac{q^g - 1}{2} \rceil\}$. Our algorithm computes the τ -adic expansion of a given scalar $\zeta \in \mathbb{Z}[\tau]$ using Algorithm 5.19 of [11] and cut this representation into d blocks of fixed size w . Each block corresponds to an element of $\mathbb{Z}[\tau]$, and we can write:

$$\zeta = \sum_{i=0}^{d-1} N_i \tau^{iw}.$$

The complete representation is obtained by replacing each N_i by its optimal $\{\tau, \tau - 1\}$ -representation obtained from a precomputed table.

We assume the following conjecture:

Conjecture 1. Let τ be a root of the characteristic polynomial of the Frobenius endomorphism of a hyperelliptic Koblitz curve. Every $\zeta \in \mathbb{Z}[\tau]$ with a finite τ -adic representation using digits in R can be represented as the sum of $O(\frac{\log \mathcal{N}(\zeta)}{\log \mathcal{N}(R)})$ numbers of the form $r_i \tau^{a_i} (\tau - 1)^{b_i}$, $r_i \in R$.

This conjecture implies that the precomputed optimal representations of the width- w blocks all have a sublinear number of terms. Numerical evidence (see, for example, [5]), suggests that Conjecture 1 holds for elliptic curves. Our belief

in the general conjecture for hyperelliptic curves is based on this evidence. Work is underway to produce supporting numerical evidence for genus 2.

Assuming Conjecture 1, we obtain the following theorem.

Theorem 4. *For fixed g and q , and assuming Conjecture 1, every $\zeta \in \mathbb{Z}[\tau]$ with a finite τ -adic expansion using digits in R can be represented as the sum of at most $O(\log \mathcal{N}(\zeta) / \log \log \mathcal{N}(\zeta))$ $\{\tau, \tau - 1\}$ -integers such that the largest power of $\tau - 1$ is $O(\log \mathcal{N}(\zeta) / \log \log \mathcal{N}(\zeta))$.*

Proof. The method used here is exactly the same as the proof of Theorem 5 of [5]: cut the τ -adic representation of ζ in $\log \log \mathcal{N}(\zeta)$ blocks, each of which is of length $O(\log \mathcal{N}(\zeta) / \log \log \mathcal{N}(\zeta))$. If each block is replaced by a representation with a sublinear number of terms, which is possible by Conjecture 1, then both the total number of terms and the highest power of $\tau - 1$ are in $O(\log \mathcal{N}(\zeta) / \log \log \mathcal{N}(\zeta))$. □

The resulting algorithm is presented in Algorithm 2. We note that the bigger the block size, the better our algorithm performs thanks to the optimality of the precomputed table. However, this has to be balanced by the increase storage cost for the larger table.

Algorithm 2. Blocking algorithm for computing $\{\tau, \tau - 1\}$ -expansions

Input: $\zeta \in \mathbb{Z}[\tau]$, block size w , precomputed table of the minimal $\{\tau, \tau - 1\}$ -expansion of every $\mu = \sum_{i=0}^{w-1} d_i \tau^i$, $d_i \in R$.

Output: List L of $\{\tau, \tau - 1\}$ -integers representing the double-base expansion of ζ .

- 1: $L \leftarrow \emptyset$
 - 2: Compute the τ -adic expansion of ζ to get an expression of the form $\zeta = \sum_{i=0}^{d-1} d_i \tau^i$.
 - 3: **for** $j = 0$ to $\lceil l/w \rceil$ **do**
 - 4: Find minimal $\{\tau, \tau - 1\}$ -expansion of $\sum_{i=0}^{w-1} d_{i+jw} \tau^i$ from the precomputed table
 - 5: Multiply by τ^{jw} and add to L
 - 6: **end for**
 - 7: **return** L
-

Our algorithm can be used for scalar multiplication as follows. To simplify the analysis we assume that the $\{\tau, \tau - 1\}$ -expansion is of the form $\zeta = \sum_{l=0}^{\max(a_i)} (\tau - 1)^l \sum_{i=0}^{\max(a_i, l)} r_{i,l} \tau^{a_i, l}$, where $\max(a_i, l)$ is the maximal power of τ that is multiplied by $(\tau - 1)^l$ in the expansion and $r_{i,j} \in R$. We then denote $r_l(\zeta) = \sum_{i=0}^{\max(a_i, l)} r_{i,l} \tau^{a_i, l}$, and thus $\zeta = \sum_{l=0}^{\max(a_i)} (\tau - 1)^l r_l(\zeta)$. The algorithm is presented in Algorithm 3.

The number of divisor additions required to compute ζD is equal to the number of terms in the expansion plus $\max(a_i) \lfloor \frac{q^g - 1}{2} \rfloor$. Provided Conjecture 1 holds, Theorem 4 implies that the total number of divisor additions is sublinear in $\log \zeta$. As before, if we assume that integer scalars of size $O(q^{ng})$ are used, then reducing the scalar modulo $(\tau^n - 1) / (\tau - 1)$ and applying the algorithm also requires a sublinear number of divisor additions.

Algorithm 3. Scalar multiplication using $\{\tau, \tau - 1\}$ -expansions

Input: $\zeta \in \mathbb{Z}[\tau]$, $D \in \text{Pic}^0(C(\mathbb{F}_q))$ **Output:** ζD

- 1: Compute $r_l(\zeta)$ for $0 \leq l \leq \max(a_i)$ such that $\zeta = \sum_{l=0}^{\max(a_i)} (\tau - 1)^l r_l(\zeta)$ using Algorithm 2.
 - 2: $D_0 \leftarrow D$
 - 3: $Q \leftarrow \text{div}[1, 0]$
 - 4: **for** $l = 0$ to $\max(a_i)$ **do**
 - 5: Compute rD_l for $r \in R$.
 - 6: $S \leftarrow r_l(m)D_l$ (τ -adic scalar multiplication using the rD_l)
 - 7: $D_{l+1} \leftarrow \tau D_l - D$ (application of $\tau - 1$)
 - 8: $Q \leftarrow Q + S$
 - 9: **end for**
 - 10: **return** Q
-

6 Conclusion and Future Work

This paper successfully generalizes ideas taken from the elliptic case to achieve improved algorithms for computing m -folds of divisor classes on hyperelliptic Koblitz curves. However, there are still a number of ways this work could be expanded.

As indicated earlier, Algorithm 1 requires provably (and unconditionally) sub-linear number of divisor additions, but only in the asymptotic sense. The O -constants involved are almost certainly too large for it to be efficient in practice. The blocking algorithm, Algorithm 3, is more promising for practical applications, but numerical experiments are required in order to determine its complexity in practice. Most importantly, we need to determine whether sufficiently short $(\tau, \tau - 1)$ -representations of all length w τ -adic numbers can be found, i.e., providing numerical evidence in support of Conjecture 1. If such short representations can be found, then the algorithm should compare favorably to the methods in [11], and a careful implementation, possibly generalizing ideas in [6] to compute $(\tau - 1)D$ efficiently, will be required. This is work in progress.

We also base our result on the hypothesis that all τ -adic expansions that we consider are finite. Although this is the case in many hyperelliptic Koblitz curves, the possibility of periods arising can be a concern in practice. We still have to come up with ways to understand those periods better, and devise efficient methods to deal with them or to avoid them completely.

Finally, we note that our double-base algorithm requires a modest amount of storage in order to achieve computational improvements. Although the pre-computed table used can be viewed as part of the domain parameters, for it does not depend on m or on the divisor class D , an efficient memory-free divisor multiplication algorithm, such as that of [1] in the case of elliptic Koblitz curves, is still left to find, as it would allow the most memory-constrained systems to enjoy this speedup as well.

Acknowledgments. The authors wish to thank the anonymous referees for their careful reading and helpful suggestions.

References

1. Avanzi, R., Dimitrov, V., Doche, C., Sica, F.: Extending Scalar Multiplication Using Double Bases. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 130–144. Springer, Heidelberg (2006)
2. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F. (eds.): Handbook of elliptic and hyperelliptic curve cryptography. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton (2006); MR2162716 (2007f:14020)
3. Dimitrov, V.S., Jullien, G.A., Miller, W.C.: An algorithm for modular exponentiation. Inform. Process. Lett. 66(3), 155–159 (1998); MR 1627991 (99d:94023)
4. Dimitrov, V.S., Imbert, L., Zakaluzny, A.: Multiplication by a constant is sublinear. In: IEEE Symposium on Computer Arithmetic 2007, pp. 261–268 (2007)
5. Dimitrov, V.S., Järvinen, K.U., Jacobson Jr., M.J., Chan, W.F., Huang, Z.: Provably sublinear point multiplication on Koblitz curves and its hardware implementation. IEEE Trans. Comput. 57(11), 1469–1481 (2008); MR2464687 (2009j:68053)
6. Doche, C., Kohel, D.R., Sica, F.: Double-Base Number System for Multi-scalar Multiplications. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 502–517. Springer, Heidelberg (2009)
7. Enge, A.: Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. Math. Comp. 71(238), 729–742 (2002); (electronic). MR 1885624 (2003b:68083)
8. Günther, C., Lange, T., Stein, A.: Speeding up the Arithmetic on Koblitz Curves of Genus Two. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 106–117. Springer, Heidelberg (2001); MR 1895585 (2003c:94024)
9. Koblitz, N.: Elliptic curve cryptosystems. Math. Comp. 48(177), 203–209 (1987); MR 866109 (88b:94017)
10. Koblitz, N.: CM-Curves with Good Cryptographic Properties. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 279–287. Springer, Heidelberg (1992)
11. Lange, T.: Efficient arithmetic on hyperelliptic curves, Ph.D. thesis, Universität-Gesamthochschule Essen, Essen, Germany (2001)
12. Miller, V.S.: Use of Elliptic Curves in Cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986)
13. Solinas, J.A.: Efficient arithmetic on Koblitz curves. Des. Codes Cryptogr. 19(2-3), 195–249 (2000)
14. Tijdeman, R.: On the maximal distance between integers composed of small primes. Compositio. Math. 28, 159–162 (1974); MR 0345917 (49 #10646)
15. Vercauteren, F.: Computing Zeta Functions of Hyperelliptic Curves Over Finite Fields of Characteristic 2. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 369–384. Springer, Heidelberg (2002)