Speeding up the Arithmetic on Koblitz Curves of Genus Two

Christian Günther
Siemens AG, Corporate Technology
Otto-Hahn-Ring 6
81730 München, Germany
christian-c.guenther@mchp.siemens.de

Tanja Lange
Institut für Geometrie
TU Braunschweig
Pockelsstr. 14
38106 Braunschweig, Germany,
ta.lange@tu-bs.de

Andreas Stein
Centre for Applied Cryptographic Research
Department of Combinatorics & Optimization
University of Waterloo
Waterloo, Ontario, N2L 3G1, CANADA
astein@math.uwaterloo.ca

January 5, 2000
CORR #2000-04

Abstract

Koblitz, Solinas, and others investigated a family of elliptic curves which admit especially fast elliptic scalar multiplication. They considered elliptic curves defined over the finite field $\mathbb{F}_2$ with base field $\mathbb{F}_{2^n}$. In this paper, we generalize their ideas to hyperelliptic curves of genus 2. Given the two hyperelliptic curves $C_a : v^2 + uv = u^5 + a\,u^2 + 1$ with $a = 0, 1$, we show how to speed up the scalar multiplication in the Jacobian $\mathbb{J}_{C_a}(\mathbb{F}_{2^n})$ by making use of the Frobenius automorphism. With some precomputations, we are able to reduce the costs of the generic double-and-add-method in the Jacobian to approximately 19 percent. If we allow a few more precomputations, we are even able to reduce the costs to about 15 percent.

1 Introduction

Public-key cryptosystems based on the discrete logarithm problem on elliptic curves over finite fields have been invented by Neal Koblitz [9] and Victor Miller [16]. Since no subexponential algorithm for solving the discrete logarithm problem (ECDLP) in the elliptic point group of a general elliptic curve is known, elliptic curve cryptosystems became a popular choice for implementations. The fastest knows attack to the ECDLP is the parallelized Pollard's rho method [18, 21, 27]. In an elliptic curve public key protocol the most important operation is the scalar multiplication by a positive integer m. That means computing mP for a point P on an elliptic curve. For example, the complexity of the ElGamal encryption scheme [4] and the Diffie-Hellmann key agreement protocol [3] on an elliptic curve both depend mostly on the complexity of the scalar multiplication. The standard method for computing m–folds in a group G is the *double-and-add-method.* If P is an element of G and m a positive integer, doublings and addings are performed with respect to the binary representation of m requiring about $\log_2(m)$ doublings and $\log_2(m)/2$ additions on average. Assuming that doubling and adding have about the same complexity, this method requires $3\log_2(m)/2$ group operations. Allowing precomputations and using memory, various techniques apply to speed up the double-and-add-method (see [8]).

In [11, 22, 14, 23], a family of elliptic curves was investigated which allows to speed up the scalar multiplication considerably with the help of the Frobenius automorphism. They considered the elliptic curves $E : u^2 + uv = v^3 + av^2 + 1$ defined over $\mathbb{F}_2$ with base field $\mathbb{F}_{2^n}$, which are called *Koblitz curves* or *anomalous binary curves (ABC curves).* As noticed in [6, 28], the attack time to these curves can be reduced by a factor of $\sqrt{2n}$ which causes one to select slightly larger secure key parameters.

Hyperelliptic curve cryptosystems have been introduced by Neal Koblitz [10] in 1989 and turned out to be a rich source of finite abelian groups for defining one-way functions. Cantor's algorithm [2] provides an effective algorithm for performing the group law in the Jacobian of a hyperelliptic curve (see also [13, 15, 24, 25] for improvements or efficient realizations). An analysis [25] shows that doubling and adding have about the same complexity. A generalization of the methods in [6, 28] shows that one can speed up the attack to hyperelliptic cryptosystems by a factor of $\sqrt{2l}$, if the curve has an automorphism of order l (see [7]).

In this paper, we generalize the ideas presented in [11, 22, 14, 12] to hyperelliptic curves of genus 2. Most of the results are easily extendable to hyperelliptic curves of arbitrary genus, but we concentrate on the following two hyperelliptic curves

$$C_a : v^2 + uv = u^5 + a\,u^2 + 1 \qquad (a = 0, 1) \ ,$$

which are defined over $\mathbb{F}_2$ and have the base field $\mathbb{F}_{2^n}$ where n is prime. These curves are generalized Koblitz curves of genus 2 and are twists of each other. Furthermore, they are the only non-supersingular curves mentioned in [10, p.147] and thus resist the Frey-Rück-attack [5]. We should remark that the curves C_a have at least an automorphism of order n. Thus, the attack to cryptosystems based on the discrete logarithm in $\mathbb{J}_{C_a}(\mathbb{F}_{2^n})$ can be sped up by a factor of $\sqrt{2n}$. As in the case of an elliptic curve, one has to adjust the size of the key space marginally. On the other side, the index calculus methods in [1, 17, 7] do not apply for curves of genus 2 (if n is reasonably large, of course).

We now proceed as follows. In Sect. 2, we introduce hyperelliptic curves and summarize some well-known facts. In Sect. 3, we develop and list the main algorithms for computing reduced τ-adic expansions and computing the scalar multiplication in the Jacobian of the hyperelliptic curve C_1. We also present a method for determining $\#\mathbb{J}_{C_a}(\mathbb{F}_{2^n})$. In Sect. 4, we list experimental data for the average length and density of the reduced τ-adic expansions and provide the factorizations of $\#\mathbb{J}_{C_a}(\mathbb{F}_{2^n})$ for prime values n. In the final section, we show how the reduced τ-adic expansion of an integer can be even shortened and give numerical evidence for the speed-up.

2 Hyperelliptic Curves

2.1 Basic Definitions

In this section we provide the basic definitions and properties of hyperelliptic curves over finite fields. We refer to [10, 15, 2, 26]. Let $\mathbb{F}$ be a finite field. A (non-singular) hyperelliptic curve of genus g is defined by the equation

$$C : v^2 + h(u)v = f(u) \quad \text{in} \quad \mathbb{F}[u, v] \ , \tag{2.1}$$

where $h(u), f(u) \in \mathbb{F}[u]$, $\deg_u(h) \leq g$, $f(u)$ monic, $\deg_u(f) = 2g + 1$, and if $y^2 + h(x)y = f(x)$ for $(x, y) \in \overline{\mathbb{F}} \times \overline{\mathbb{F}}$, then $2y + h(x) \neq 0 \lor h'(x)y - f'(x) \neq 0$. Let $\mathbb{K}$ be a subfield of $\overline{\mathbb{F}}$ containing $\mathbb{F}$. The set of $\mathbb{K}$–points P on C is given by $C(\mathbb{K}) = \{(x, y) \in \mathbb{K}^2 \mid y^2 + h(x)y = f(x)\} \cup \{\infty\}$, where ∞ denotes the point at infinity. For a $\mathbb{K}$–point $P = (x, y) \in \mathbb{K}^2$, the *opposite* $\tilde{P}$ of P is immediately given by $\tilde{P} = (x, -y - h(x))$. For $P = \infty$ define $\tilde{P} = \infty$. A *divisor* on C is a finite formal sum $D = \sum_P m_P P$, where m_P are integers that are 0 for almost all P. Then, the degree of D is defined by $\deg D = \sum_P m_P$. D is said to be *defined over* $\mathbb{K}$, if [1] $D^\sigma = \sum_P m_P P^\sigma = D$ for all $\sigma \in$ Aut $(\overline{\mathbb{F}}/\mathbb{K})$. The set $\mathbb{D}_C(\mathbb{K})$ of divisors of C defined over $\mathbb{K}$ forms an additive group which contains the finite subgroup $\mathbb{D}_C^0(\mathbb{K})$ of all degree zero divisors of $\mathbb{D}$ defined over $\mathbb{K}$. The divisor of a polynomial $G(u, v) \in \overline{\mathbb{F}}[u, v]$ is defined by $\mathrm{div}(G(u, v)) = \sum_P \mathrm{ord}_P(G)P - \sum_P \mathrm{ord}_P(G)\infty$, where $\mathrm{ord}_P(G)$ is the order of vanishing of $G(u, v)$ at P. Now, the divisor of a rational function $G(u, v)/H(u, v)$ is called a principal divisor and is defined by $\mathrm{div}(G(u, v)/H(u, v)) = \mathrm{div}(G(u, v)) - \mathrm{div}(H(u, v))$. We denote by $\mathbb{P}_C(\mathbb{K})$ the group of principal divisors. Since every principal divisor has degree 0, $\mathbb{P}_C(\mathbb{K})$ is a subgroup of $\mathbb{D}_C^0(\mathbb{K})$. Finally, the Jacobian of C over $\mathbb{K}$ is given by $\mathbb{J}_C(\mathbb{K}) = \mathbb{D}_C^0(\mathbb{K})/\mathbb{P}_C(\mathbb{K})$. It is well-known (see for instance [19, 20]) that each divisor in $\mathbb{D}_C^0(\mathbb{K})$ is equivalent to a unique reduced divisor. Thus, every element of the Jacobian can be uniquely represented by a pair of polynomials $[a(u), b(u)]$, where $a(u), b(u) \in \mathbb{K}[u]$ such that $a(u)$ is monic, $\deg b(u) < \deg a(u)$, and $a(u)$ divides $b(u)^2 + b(u)h(u) - f(u)$. We notice that operations in the Jacobian can be performed by using the arithmetic in $\mathbb{K}[u]$. Without explaining the algorithms here, we mention that there exists effective method to add two elements of the Jacobian which is known as Cantor's algorithm. For details, we refer to [2, 10, 15, 25, 20]. The generic operation need $17g^2 + O(g)$ operations in $\mathbb{K}$

[1] P^σ denotes $(\sigma(x), \sigma(y))$, if $P = (x, y) \in \mathbb{K}^2$, and ∞, if $P = \infty$.

whereas doubling needs $16g^2 + O(g)$ operations in $\mathbb{K}$. [2] So, we can assume that both operations have roughly the same complexity. It is important to note that inversion is basically for free, since the opposite of $D = [a(u), b(u)]$ is given by $\mathrm{div}[a(u), -h(u) - b(u)]$.

2.2 Frobenius Automorphism

In this section, we assume that $C : v^2 + h(u)v = f(u)$ is a hyperelliptic curve of genus g defined over the finite field $\mathbb{F} = \mathbb{F}_q$ of q elements. We let $\mathbb{K} = \mathbb{F}_{q^n}$ for a positive integer n. The Frobenius automorphism $\phi : \overline{\mathbb{F}}_q \longrightarrow \overline{\mathbb{F}}_q$, $x \longmapsto x^q$ induces an endomorphism

$$\begin{array}{rcl} \phi : \quad \mathbb{J}_C(\overline{\mathbb{F}}_q) & \longrightarrow & \mathbb{J}_C(\overline{\mathbb{F}}_q) \\ \left(\sum_P m_P P\right) \mod \mathbb{P}_C(\overline{\mathbb{F}}_q) & \longmapsto & \left(\sum_P m_P P^{\phi}\right) \mod \mathbb{P}_C(\overline{\mathbb{F}}_q) \ , \end{array} \tag{2.2}$$

where $P^{\phi} = (x^q, y^q)$, if $P = (x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$, and $P^{\phi} = \infty$, if $P = \infty$. For a divisor $D = \sum_P m_P P$ of C define D^{ϕ} to be $\sum_P m_P P^{\phi}$.

An important property of the Frobenius of such hyperelliptic curves is that if $D = [a(u), b(u)]$ is a reduced divisor, then $D^{\phi} = [a(u)^{\phi},\ b(u)^{\phi}]$. Thus, if $a(u) = \sum_{i=0}^{k} a_i u^i \in \mathbb{K}[u]$ and $b(u) = \sum b_i u^i \in \mathbb{K}[u]$, then $a^{\phi}(u) = \sum_{i=0}^{k} a_i^q u^i$ and $b^{\phi}(u) = \sum_{i=0}^{k} b_i^q u^i$. The computation of D^{ϕ} then reduces to at most $2g$ operations in $\mathbb{K}$. The practical meaning of this observation is that if we use normal basis representation for elements in $\mathbb{F}_{2^n}$, then $a^{\phi}(u)$ and $b^{\phi}(u)$ can be determined by simply shifting the normal basis representation of each coefficient a_i and b_i in order to compute D^{ϕ}. The complexity is therefore at most $2g$ cyclic shifts. These shift operations are basically "for free" when compared to the more expensive group operation in the Jacobian.

3 Algorithms for $v^2 + uv = u^5 + a\,u^2 + 1$

For the remainder of the paper, we consider the curves $C_a : v^2 + uv = u^5 + a\,u^2 + 1$ with $a = 0, 1$ which are defined over $\mathbb{F}_2$. From [10], we know that the characteristic polynomial of the Frobenius of the curve $C_1 : v^2 + uv = u^5 + u^2 + 1$ is given by

$$\varphi(T) = T^4 - T^3 - 2T + 4 \ . \tag{3.3}$$

[2]We remark that there exist even faster methods if the characteristic of $\mathbb{K}$ is 2 and if we use normal basis representation for elements in $\mathbb{K}$.

It follows that

$$4D \equiv -\phi^4(D) + \phi^3(D) + 2\phi(D) \mod \mathbb{P}_{C_1}(\overline{\mathbb{F}}_2)$$

for all divisors $D \in \mathbb{D}^0_{C_1}(\overline{\mathbb{F}}_2)$. The characteristic equation $\varphi(T) = 0$ has four solutions

$$\tau_{1/2} = (\mu_1 \pm i\sqrt{4-\mu_1})/2 \quad , \quad \tau_{3/4} = (\mu_2 \pm i\sqrt{4-\mu_2})/2 \ ,$$

where $\mu_{1/2} = (1 \pm \sqrt{17})/2$. We put $\tau = \tau_1$ and can regard τ as the element ϕ in the endomorphism ring of $\mathbb{J}_{C_1}(\overline{\mathbb{F}}_2)$.

Now, the curve $C_0 : v^2 + uv = u^5 + 1$ has the characteristic equation $T^4 + T^3 + 2T + 4 = 0$. Thus, the roots of this equation are simply given by $-\tau_1, -\tau_2, -\tau_3, -\tau_4$, and the curve C_0 is just the twist of C_1. It therefore suffices to consider C_1. Analogous results hold true for the curve C_0 with some slight modifications. In particular, $\#\mathbb{J}_{C_0}(\mathbb{F}_{2^n})$ differs from $\#\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$ only for odd n (see Sect. 3.6).

3.1 Computing τ-adic Expansions

We are interested in expansions like $11 = -\tau^7 + \tau^4 - 2\tau^2 + 3$, which enable us to compute $11D$ by $11D = -\phi^7(D) + \phi^4(D) - 2\phi^2(D) + 3D$ for $D \in \mathbb{D}^0_{C_1}(\overline{\mathbb{F}}_2)$. More generally, we are interested in expansions of the form

$$m = \sum_{i=0}^{l-1} c_i \tau^i \quad (m \in \mathbb{Z}[\tau],\, c_i \in R,\, l \geq 1) \ , \tag{3.4}$$

where R is a suitable set for the coefficients c_i. First, we consider $R = \{0, \pm 1, \pm 2, \pm 3\}$. In Sect. 5, we will vary the set R. Since τ is a root of (3.3), an element $m = a + b\tau + c\tau^2 + d\tau^3 \in \mathbb{Z}[\tau]$ with integers a, b, c, d is divisible by τ if and only if $4 \mid a$ in $\mathbb{Z}$. We can see this as follows. First, suppose that $\tau \mid m$. Then there exist integers $\overline{a}, \overline{b}, \overline{c}, \overline{d}$ such that

$$\begin{aligned} m &= \tau(\overline{a} + \overline{b}\tau + \overline{c}\tau^2 + \overline{d}\tau^3) = \overline{a}\tau + \overline{b}\tau^2 + \overline{c}\tau^3 + \overline{d}(\tau^3 + 2\tau - 4) \\ &= -4\overline{d} + (\overline{a} + 2\overline{d})\tau + \overline{b}\tau^2 + (\overline{c} + \overline{d})\tau^3 \ . \end{aligned}$$

Since $m = a + b\tau + c\tau^2 + d\tau^3$, we conclude that $4 \mid a$. If we assume that $4 \mid a$, then there exists an integer $\overline{a} \in \mathbb{Z}$ such that

$$\begin{aligned} m &= 4\overline{a} + b\tau + c\tau^2 + d\tau^3 = (-\tau^4 + \tau^3 + 2\tau)\overline{a} + b\tau + c\tau^2 + d\tau^3 \\ &= \tau\left((2\overline{a} + b) + c\tau + (\overline{a} + d)\tau^2 - \overline{a}\tau^3\right) \ . \end{aligned}$$

Thus, $\tau \mid m$. Therefore, there is exactly one $u \in \{0, 1, 2, 3\}$ such that $\tau \mid m - u$ and

$$m - u = \tau\Big(\Big(\frac{a-u}{2} + b\Big) + c\tau + \Big(\frac{a-u}{4} + d\Big)\tau^2 - \frac{a-u}{4}\tau^3\Big) \ . \tag{3.5}$$

With $R = \{0, \pm 1, \pm 2, \pm 3\}$ we are able to realize the strategy "*at least one of four consecutive coefficients is zero*" when determining the c_i's. The basic algorithm for computing τ-adic expansions of $m = a + b\tau + c\tau^2 + d\tau^2 \in \mathbb{Z}[\tau]$ is to choose an $u \in R$ such that $4 \mid m - u$, to divide $m - u$ by τ and then to repeat these two steps with the new, replaced $m = ((a-u)/2 + b) + c\tau + ((a-u)/4 + d)\tau^2 - ((a-u)/4)\tau^3$, see (3.5), until m will be zero. Then the sequence of those u's will be the sequence of the coefficients $c_0, \ldots, c_{l-1} \in R$ we were looking for. In (3.5) you can see what we have to do for realizing the strategy "*at least one of four consecutive coefficients is zero*":

1.) If $4 \mid a$, then $\tau \mid m$ and we clearly use $u = 0$.

2.) If $4 \nmid a$, then since $R = \{0, \pm 1, \pm 2, \pm 3\}$ we have exactly two choices for u and we can try to make one of the subsequent a's divisible by 4:

   a.) If $2 \mid b$, then there is exactly one $u \in R$ such that $4 \mid a - u$ and $4 \mid ((a-u)/2 + b)$, namely

u	$a \mod 8$					
$b \mod 4$	1	2	3	5	6	7
0	1	2	3	-3	-2	-1
2	-3	-2	-1	1	2	3

   Using these values for u, the actual u is non zero but the next one will be zero.

   b.) If $2 \nmid b$, then we cannot make both $(a - u)$ and $((a - u)/2 + b)$ be divisible by 4. And we have no influence on the following b, since this will be just c. But there is exactly one $u \in R$ such that $4 \mid (a - u)$ and $2 \mid ((a - u)/4 + d)$, namely

u	$a \mod 8$					
$d \mod 2$	1	2	3	5	6	7
0	1	2	3	-3	-2	-1
1	-3	-2	-1	1	2	3

Now, the number $(a-u)/4+d$ is even, which enables us to force the third successor of the actual a at the latest to be divisible by 4, see (3.5) and a.) in 2.).

This strategy produces expansions $m=\sum_{i=0}^{l-1} c_i\tau^i$, $c_i \in R = \{0, \pm 1, \pm 2, \pm 3\}$, $l \geq 1$, with

$$c_i c_{i+1} c_{i+2} c_{i+3} = 0 \quad (i \in \{0, \dots, l-4\}), \tag{3.6}$$

and leads to the following

Algorithm 3.1. *(Computing τ-adic Expansions)*

`INPUT`: $m = a + b\tau + c\tau^2 + d\tau^3 \in \mathbb{Z}[\tau]$

`OUTPUT`: $c_0, \dots, c_{l-1} \in R = \{0, \pm 1, \pm 2, \pm 3\}$ *with* $m = \sum_{i=0}^{l-1} c_i\tau^i$.

1.) $i \leftarrow 0$;

2.) `While` ($a \neq 0$ `or` $b \neq 0$ `or` $c \neq 0$ `or` $d \neq 0$)

a.) $u \leftarrow a \pmod 4$;

b.) `If` ($u \neq 0$))

`If` (($b \mod 4 = 0$ `and` $a \mod 8 > 4$) `or`
($b \mod 4 = 2$ `and` $a \mod 8 < 4$) `or`
($b \mod 2 = 1$ `and` $a \mod 8 > 4$ `and` $d \mod 2 = 0$) `or`
($b \mod 2 = 1$ `and` $a \mod 8 < 4$ `and` $d \mod 2 = 1$))
$u \leftarrow u - 4$

c.) $c_i \leftarrow u$;

d.) $v \leftarrow (a-u)/4$; $a \leftarrow 2v + b$; $b \leftarrow c$; $c \leftarrow v + d$; $d \leftarrow -v$;

e.) $i \leftarrow i + 1$;

f.) `Output(` c_i `)` .

The finiteness of the algorithm can be derived from the following considerations. With the complex absolute value the following triangle inequality holds for elements of $\alpha, \beta \in \mathbb{Q}[\tau]$:

$$|\alpha + \beta| \leq |\alpha| + |\beta| \ .$$

Therefore in the process of computing the expansion, the absolute value of the remaining element decreases according to

$$\sqrt{2}\,|\alpha_{new}| = |\alpha + \beta| \leq |\alpha| + |\beta| \leq |\alpha| + 3 \ ,$$

where $\alpha = a + b\tau + c\tau^2 + d\tau^3$ is the element before it is made divisible by τ, $\beta \in R$ is the remainder and $\alpha_{new} = (\alpha + \beta)/\tau$ is the new element. So for $|\alpha| > 8$ we have $|\alpha| > |\alpha_{new}|$. Our experiments show that the expansion is always finite. However, we were unable to close this final gap so far.

Unfortunately, the above algorithm does not produce expansions $m = \sum c_i\tau^i$ that have the minimal number of nonzero coefficients among all expansions $m = \sum c_i\tau^i$ with $c_i \in \{0, \pm1, \pm2, \pm3\}$. Assuming the expansion to be finite we will derive bounds on the length of it (cf. [22]). By the length of an element of $\mathbb{Z}[\tau]$ we mean the length of its τ-adic representation. Let $V_{max}(k)$ be the largest absolute value occurring among all length-k elements of $\mathbb{Z}[\tau]$. We have $\sqrt{2}\,V_{max}(k) \leq V_{max}(k+1)$, as if α is a length-k element of maximal absolute value, then $\tau\alpha$ is an element of length $k+1$ and absolute value $\sqrt{2}\,|\alpha|$, i.e. $V_{max}(k)$ is the largest absolute value occurring among all elements $\alpha \in \mathbb{Z}[\tau]$ of length at most k.

If $c > e$ then we can show that

$$V_{max}(c) \leq 2^{e/2}\,V_{max}(c - e) + V_{max}(e) \ . \tag{3.7}$$

If $l > d$, then we obtain

$$V_{max}(l) < \frac{V_{max}(d)}{2^{d/2} - 1}\,2^{l/2} \ .$$

We now let V_{min} denote the smallest absolute value occurring among all length-k elements of $\mathbb{Z}[\tau]$. If $c > e$, then $V_{min}(c) \geq 2^{e/2}V_{min}(c-e) - V_{max}(e)$. For $l > 2d$ we even have $V_{min}(l) > (V_{min}(d) - \frac{V_{max}(d)}{2^{d/2}-1}) \cdot 2^{(l-d)/2}$. The following theorem holds.

Theorem 3.2. *Let $l > 2d$, and let α be a length-l element of $\mathbb{Z}[\tau]$. Then*

$$\left(V_{min}(d) - \frac{V_{max}(d)}{2^{d/2} - 1}\right) \cdot 2^{(l-d)/2} < |\alpha| < \frac{V_{max}(d)}{2^{d/2} - 1} \cdot 2^{l/2} \ .$$

So the length of the representation is approximately $2\log_2(|\alpha|)$, as

$$2|\,\alpha\,| - 2\log_2\left(\frac{V_{max}(d)}{2^{d/2}-1}\right) \;<\; l \;<\; 2\,|\alpha\,| \,+\, d \,-\, 2\,\log_2\left(V_{min}(d) \,-\, \frac{V_{max}(d)}{2^{d/2}-1}\right)\;,$$

if $V_{min}(d) > V_{max}(d)/(2^{d/2}-1)$. But, this inequality is satisfied for sufficiently large values of d. The expected length l of an integer $m = \sum_{i=0}^{l-1} c_i\tau^i$ is $2\log_2|m|$, which is about twice as long as the binary expansion $m = \pm\sum b_i 2^i$, $b_i \in \{0,1\}$, of m. We will show later how to reduce the length of the τ-adic representation.

3.2 Dividing Integers by $\tau^{\mathbf{n}} - 1$ in $\mathbb{Z}[\tau]$

Let $\sum_{i=0}^{l_1-1} c_i\tau^i$ and $\sum_{i=0}^{l_2-1} d_i\tau^i$, be two elements in $\mathbb{Z}[\tau]$ that are congruent modulo τ^n-1 for some positive integer n, i.e.

$$\sum_{i=0}^{l_1-1} c_i\tau^i - \sum_{i=0}^{l_2-1} d_i\tau^i \;\in\; (\tau^n-1)\,\mathbb{Z}[\tau]\;.$$

The corresponding endomorphisms $\sum_{i=0}^{l_1-1} c_i\phi^i$, $\sum_{i=0}^{l_2-1} d_i\phi^i$ in $\mathrm{End}(\mathbb{J}_{C_1}(\mathbb{F}_{2^n}))$ are the same, since

$$\sum_{i=0}^{l_1-1} c_i\phi^i - \sum_{i=0}^{l_2-1} d_i\phi^i \;\in\; (\phi^n-1)\,\mathbb{Z}[\phi] \;\subset\; \mathrm{End}(\mathbb{J}_{C_1}(\mathbb{F}_{2^n}))$$

and $\phi^n - 1 = 0$ in $\mathrm{End}(\mathbb{J}_{C_1}(\mathbb{F}_{2^n}))$. Therefore, in order to obtain short representations $[m] = \sum_{i=0}^{l-1} c_i\phi^i$ of the multiplication-by-m-map

$$\begin{array}{rccc} [m]\;: & \mathbb{J}_{C_1}(\mathbb{F}_{2^n}) & \longrightarrow & \mathbb{J}_{C_1}(\mathbb{F}_{2^n}) \\ & D \mod \mathbb{P}_{C_1}(\mathbb{F}_{2^n}) & \longmapsto & mD \mod \mathbb{P}_{C_1}(\mathbb{F}_{2^n})\;, \end{array} \tag{3.8}$$

we look for an element $M \in \mathbb{Z}[\tau]$ such that $M \equiv m \mod \tau^n - 1$ and the τ-adic expansion of M is as short as possible. In other words, we look for elements M and z in $\mathbb{Z}[\tau]$ such that $m = z(\tau^n-1) + M$ and $|M|$ is as small as possible.

Theorem 3.3. *For any nonzero integer m and positive integer n, there exists an element $M \in \mathbb{Z}[\tau]$ such that*

1.) $m \equiv M \mod \tau^n - 1$,

2.) $2\log_2|M| < n+5$.

Proof. Let $q = m/(\tau^n - 1) \in \mathbb{Q}(\tau)$. Then there exist q_0, q_1,q_2, q_3 in $\mathbb{Q}$ such that $q = \sum_{i=0}^{3} q_i\tau^i$. Choose z_0, z_1, z_2, $z_3 \in \mathbb{Z}$ such that $\mid q_i - z_i \mid \leq \frac{1}{2}$. Let z and M be the elements $z = \sum_{i=0}^{3} z_i\tau^i$ and $M = m - z(\tau^n - 1)$. Then we have $m \equiv M \mod \tau^n - 1$. We obtain

$$\begin{aligned} \left|\frac{m}{\tau^n - 1} - z\right|^2 &= |q - z|^2 = \left|\sum_{i=0}^{3}(q_i - z_i)\tau^i\right|^2 \\ &\leq \left(\frac{1}{2}\sum_{i=0}^{3}\sqrt{2}^i\right)^2 \\ &= \left(\frac{3}{2}(1+\sqrt{2})\right)^2 < 14 \ . \end{aligned}$$

It follows that

$$|M|^2 = |m - z(\tau^n - 1)|^2 < 14 \cdot |\tau^n - 1| \leq 14 \cdot (2^{n/2} + 1)^2 \ ,$$

and hence

$$2\log_2|M| < \log_2(14) + 2\log_2(2^{n/2} + 1) < n + 5 \ .$$

□

For given $m \in \mathbb{Z} - \{0\}$ and n in $\mathbb{N}$, we are now able to compute an element $M = \sum_{i=0}^{3} M_i\tau^i$, $M_i \in \mathbb{Z}$, satisfying $m \equiv M \mod \tau^n - 1$ which has a τ-adic expansion $M = \sum_{i=0}^{l-1} c_i\tau^i$ where l is in the order of n. We call this representation the *reduced τ-adic expansion* of m. In the endomorphism ring $\mathrm{End}(\mathbb{J}_{C_1}(\mathbb{F}_{2^n}))$, we obtain for the multiplication-by-m map that $[m] = \sum_{i=0}^{l-1} c_i\phi^i$. The algorithm to compute M from m is along the lines of the proof of Theorem 3.3. We therefore omit it. We remark here that we need to be able to find a representation of $\tau^n - 1$ as $\tau^n - 1 = a + b\tau + c\tau^2 + d\tau^3$ with integers a, b, c, d. Furthermore, we need to be able to compute multiplicative inverses in $\mathbb{Z}[\tau]$. The next two sections will solve these problems.

3.3 Representing $\tau^{\mathbf{n}} - 1$ by $\mathbf{a} + \mathbf{b}\tau + \mathbf{c}\tau^2 + \mathbf{d}\tau^3$

To compute a, b, c, $d \in \mathbb{Z}$ such that $\tau^n - 1 = a + b\tau + c\tau^2 + d\tau^3$ is no difficult task. Let $n \in \mathbb{N}$. Suppose that

$$\tau^{n-1} = a_{n-1} + b_{n-1}\tau + c_{n-1}\tau^2 + d_{n-1}\tau^3$$

for unique integers a_{n-1}, b_{n-1}, c_{n-1}, d_{n-1}, then

$$\begin{aligned}\tau^n &= a_{n-1}\tau + b_{n-1}\tau^2 + c_{n-1}\tau^3 + d_{n-1}\tau^4 \\ &= -4d_{n-1} + (a_{n-1} + 2d_{n-1})\tau + b_{n-1}\tau^2 + (c_{n-1} + d_{n-1})\tau^3 \ ,\end{aligned}$$

since $\tau^4 = -4 + 2\tau + \tau^3$, and hence

$$\tau^n - 1 = -(4d_{n-1} + 1) + (a_{n-1} + 2d_{n-1})\tau + b_{n-1}\tau^2 + (c_{n-1} + d_{n-1})\tau^3.$$

Starting with $\tau^0 = 1$, we can compute the integers a, b, c, d iteratively:

Algorithm 3.4. *(Representing $\tau^n - 1$ by $a + b\tau + c\tau^2 + d\tau^3$)*

`INPUT`: *A positive integer n.*

`OUTPUT`: *Integers a, b, c, d such that $\tau^n - 1 = a + b\tau + c\tau^2 + d\tau^3$.*

1.) $a \leftarrow 1$; $b \leftarrow 0$; $c \leftarrow 0$; $d \leftarrow 0$; $k \leftarrow 1$;

2.) `While` ($k \leq n$)

- *a.)* $a_{old} \leftarrow a$; $b_{old} \leftarrow b$; $c_{old} \leftarrow c$; $d_{old} \leftarrow d$;
- *b.)* $a \leftarrow -4d_{old}$;;
- *c.)* $b \leftarrow a_{old} + 2d_{old}$;
- *d.)* $c \leftarrow b_{old}$;
- *e.)* $d \leftarrow c_{old} + d_{old}$;
- *f.)* $k \leftarrow k + 1$;

3.) $a \leftarrow a - 1$;

4.) `Output`(a, b, c, d) ;

3.4 Inversion of Elements $\mathbf{a + b\tau + c\tau^2 + d\tau^3}$

We show how to compute the multiplicative inverse of $M = a + b\tau + c\tau^2 + d\tau^3$ in $\mathbb{Z}[\tau]$. This can be established as follows. We compute the extended Euclidean algorithm of $R_0(T) = T^4 - T^3 - 2T + 4$ and $R_1 = a + bT + cT^2 + dT^3$. Since $\mathbb{Q}[T]$ is a Euclidean domain with respect to the degree map, there exist unique polynomials $V(T), U(T), G(T) \in \mathbb{Q}[T]$ such that

$$G(T) = \gcd(R_0(T), R_1(T)) = V(T)\, R_0(T) \;+\; U(T)\, R_1(T) \ .$$

Since $R_0(T)$ is irreducible in $\mathbb{Q}[T]$ and $\deg R_1(T) < \deg R_0(T)$, we must have that $G(T) = \beta \in \mathbb{Q}$. If we insert τ for T and use that $R_0(\tau) = 0$, we obtain

$$\beta = V(\tau)\, R_0(\tau) + U(\tau)\, R_1(\tau) = U\,(\tau)\, R_1(\tau) \ .$$

Hence,

$$(a + b\tau + c\tau^2 + d\tau^3)^{-1} = U(\tau)/\beta \ .$$

3.5 Computing m-folds of Divisor Classes Using τ-adic Expansions

We now present our main algorithm for computing m-folds of divisor classes of the genus 2 curve $C_1 \,:\, v^2 + uv = u^5 + u^2 + 1$ with base field F_{2^n}. Let $D = \mathrm{div}(a(u), b(u))$ be the unique representation of an element of the Jacobian $\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$, where $a(u) = a_0 + a_1 u + u^2$ and $b(u) = b_0 + b_1 u$ with coefficients $a_0, a_1, b_0, b_1 \in \mathbb{F}_{2^n}$. Let the coefficients a_0, a_1, b_0, b_1 be represented with respect to a normal basis $B = \{\alpha, \alpha^2, \alpha^{2^2}, \ldots, \alpha^{2^{n-1}}\}$ of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$, i.e.

$$a_k = \sum_{i=0}^{n-1} a_{ki} \alpha^{2^i} \quad , \quad b_k = \sum_{i=0}^{n-1} b_{ki} \alpha^{2^i} \qquad (\, a_{ki}, b_{ki} \in \mathbb{F}_2 \ , \ k \in \{0, 1\}\,) \ .$$

Recall that

$$\phi^4(D) - \phi^3(D) - 2\phi(D) + 4D \ \in \ \mathbb{P}_{C_1}(\overline{\mathbb{F}}_2)$$

and that every expansion $m = \sum_{i=0}^{l-1} c_i \tau^i$, with integers m, c_i, yields a corresponding representation $[m] = \sum_{i=0}^{l-1} c_i \phi^i$ of the multiplication-by-m-map. Working in the finite group $\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$, we can additionally exploit the fact that $\phi^n(D) = D$ for all $D \in \mathbb{D}^0_{C_1}(\mathbb{F}_{2^n})$. By our previous considerations, we can assume that the we already computed the reduced τ-adic representation of m, i.e. we computed $c_0, \ldots, c_{l-1} \in R$ such that $m \equiv \sum_{i=0}^{l-1} c_i \tau^i \pmod{\tau^n - 1}$.

Algorithm 3.5. *(Computing Scalar Multiples of Divisor Classes)*

`INPUT:` $c_0, \ldots, c_{l-1} \in \{0, \pm 1, \pm 2, \pm 3\}$ *with* $m \equiv \sum_{i=0}^{l-1} c_i \tau^i \pmod{\tau^n - 1}$.
and $a_0, a_1, b_0, b_1 \in \mathbb{F}_{2^n}$ *representing a divisor class* $[D] \in \mathbb{J}_{C_1}(\mathbb{F}_{2^n})$.
`OUTPUT:` $s_0, s_1, t_0, t_1 \in \mathbb{F}_{2^n}$ *representing the divisor class* $m[D] \in \mathbb{J}_{C_1}(\mathbb{F}_{2^n})$.

1.) *Precompute the divisors* $2D$, $3D$.

2.) *Initialize* $H = \mathrm{div}(s(u), t(u))$ *with* $s(u) = 1$, $t(u) = 0$ *representing the principal class.*

3.) `For` i `from` $l-1$ `downto` 0 `do`

a.) $H \leftarrow \phi(H)$;

b.) `If` ($c_i \neq 0$) $H \leftarrow H + c_i D$;

4.) `Output(`H`)` ; `/*` *i.e.* `output(`s_0, s_1, t_0, t_1`)` `*/`

Note that the operation $H = \phi(H)$ is nothing else than cyclic shifting of at most 4 coefficients s_0, s_1, t_0, t_1 of $s(u)$ and $t(u)$, if s_0, s_1, t_0, t_1 are represented with respect to a normal basis.

In the last paragraphs we will give some statistics on the length and the density of the τ-adic expansions obtained in step 3) of this algorithm. We will also provide some data on how to shorten the expansions.

3.6 Computing the Number of Divisor Classes

In this paragraph, we follow the lines of [10] and show how to compute the positive number $N_n = \#\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$. We know that

$$\begin{aligned} N_n &= \#\mathbb{J}_{C_1}(\mathbb{F}_{2^n}) = N(1 - \tau_1^n) = \prod_{i=1}^{4}(1 - \tau_i^n) \\ &= \Big((1 + 2^n) - (\tau_1^n + \tau_2^n)\Big)\Big((1 + 2^n) - (\tau_3^n + \tau_4^n)\Big) \ , \end{aligned} \tag{3.9}$$

where N denotes the usual norm map for $\mathbb{Q}(\tau_1)/\mathbb{Q}$. An immediate formula for N_n appears to be hard to develop. A possible solution is to compute $\tau_1^n + \tau_2^n$ and $\tau_3^n + \tau_4^n$. Since τ_1

(and each other τ_i) is an algebraic integer and $\tau_1^n + \tau_2^n = \tau_1^n + \overline{\tau}_1^n = \tau_1^n + (\mu_1 - \tau_1)^n \in \mathbb{Q}(\tau_1) \cap \mathbb{R} = \mathbb{Q}(\mu_1)$, there are, for all $n \in \mathbb{N}$, integers A_n and B_n such that

$$\tau_1^n + \tau_2^n = A_n + \mu_1 B_n \ , \tag{3.10}$$

and we can try to determine A_n and B_n recursively. For $n \geq 2$ we get

$$\tau_1^n + \tau_2^n = (4B_{n-1} - 2A_{n-2}) + \mu_1(A_{n-1} + B_{n-1} - 2B_{n-2}) \ .$$

Equating coefficients leads to the following definition

$$\begin{aligned} &A_0 = 2,\ A_1 = 0,\ A_n = 4B_{n-1} - 2A_{n-2} \ \text{ for } \ n \geq 2, \\ &B_0 = 0,\ B_1 = 1,\ B_n = A_{n-1} + B_{n-1} - 2B_{n-2} \ \text{ for } \ n \geq 2, \end{aligned}$$

in order to force

$$\tau_1^n + \tau_2^n = A_n + \mu_1 B_n \quad \textit{and} \quad \tau_3^n + \tau_4^n = A_n + \mu_2 B_n \quad (n \geq 0) \ .$$

By using these formulas, we can easily compute N_n by

$$\begin{aligned} N_n &= \big((1+2^n) - (A_n + \mu_1 B_n)\big)\big((1+2^n) - (A_n + \mu_2 B_n)\big) \\ &= (1+2^n)^2 - (2A_n + B_n)(1+2^n) + (A_n^2 + A_n B_n - 4B_n^2). \end{aligned}$$

Notice that we can determine $\#\mathbb{J}_{C_0}(\mathbb{F}_{2^n})$ in a similar fashion by

$$\#\mathbb{J}_{C_0}(\mathbb{F}_{2^n}) = (1+2^n)^2 - (-1)^n(2A_n + B_n)(1+2^n) + (A_n^2 + A_n B_n - 4B_n^2) \ ,$$

since the roots of the characteristic polynomial of C_0 are $-\tau_1, -\tau_2, -\tau_3, -\tau_4$.

Finally, we mention here, that $N_n \sim 2^{2n}$ as a result of the considerations above, where we explicitly used the Theorem of Weil.

4 Experimental Results

This section contains three tables. Table 1 describes the length and the density of reduced τ-adic expansions For each prime $n \in \{61, \ldots, 113\}$, we generated 10000 random integers m in the range $0 < m < \#\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$. We computed the reduced τ-adic representation of each $m = \sum_{i=0}^{l-1} c_i \tau^i$ of length l. If d denotes the number of the nonzero coefficients c_i, the quotient l/d is its density.

Table 1: Average Length and Density

n	average length	average density	n	average length	average density
61	62.38	0.5460	97	98.34	0.5437
67	68.36	0.5458	101	102.36	0.5433
71	72.38	0.5455	103	104.31	0.5429
73	74.35	0.5449	107	108.33	0.5434
79	80.33	0.5445	109	110.34	0.5424
83	84.35	0.5440	113	114.35	0.5427
89	90.32	0.5441			

The value $n+\frac{4}{3}$ seems to be a good approximation for the expected length l of a reduced τ-adic expansion. The average density for degrees n in the range from 61 to 113 is about 54.5 percent, so that the expected number of nonzero coefficients c_i is approximately $\frac{545}{1000}(n+\frac{4}{3}) \sim \frac{5}{9}n$.

Therefore, Algorithm 3.5 for computing multiples $m[D]$ of divisor classes $[D] \in \mathbb{J}_{C_1}(\mathbb{F}_{2^n})$ needs about $\frac{5}{9}n$ additions of reduced divisors, while the shift operations are essentially for free. The double-and-add-method for $\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$ needs about $2n$ doublings and n additions of reduced divisors, so that the τ-adic method reduces the costs for multiplying divisor classes to roughly

$$\frac{5}{9}n/3n \sim 19\%$$

of the costs of the double-and-add-method.

Table 2 and 3, resp., list the factorizations of $\#J_{C_1}(\mathbb{F}_{2^n})$ and $\#J_{C_0}(\mathbb{F}_{2^n})$ for prime values of n in the range between 61 and 113.

Table 2: Computing the Cardinality of the Jacobian $\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$

n	$\#\mathbb{J}_{C_1}(\mathbb{F}_{2^n})$
61	$5316911976894487061973100640561324954 =$
	$2 \cdot 2658455988447243530986550320280662477$
67	$21778071481105140023832236795388122729642 =$
	$2 \cdot 3217 \cdot 3384841697405212935006564624710619013$
71	$5575186299560430202994122000844046836505866 =$
	$2 \cdot 454969 \cdot 447728273 \cdot 805164709 \cdot 16996062957750093401$
73	$89202980790795799816393385454503895169367738 =$
	$2 \cdot 29487329 \cdot 95930761 \cdot 118654201 \cdot 132884071749443674301$
79	$365375409332917774587636484565802686769448765898 =$
	$2 \cdot 8059 \cdot 1994119 \cdot 8949518819549513 \cdot 1270215495254265193313$
83	$93536104789224306098427384543147920201461688362538 =$
	$2 \cdot 228251 \cdot 1344767 \cdot 15183347701 \cdot 10035107170580262465826364557$
89	$383123885216493271959483132021014047072341682130661434 =$
	$2 \cdot 179 \cdot 10859 \cdot 340693 \cdot 1309013 \cdot 859598867342557 \cdot 257077083193572379769$
97	$25108406941546737996390354885625124943376439570684227477754 =$
	$2 \cdot 389 \cdot 1747 \cdot 18473392463868826910318794676754071940716909907019619$
101	$6427752177035957949506966525786377643809064101189343179038554 =$
	$2 \cdot 16053143 \cdot 11100831153947 \cdot 22216548397721 \cdot 811777425582909977125409897$
103	$102844034832575383397207943835010553634640254575820398436691978 =$
	$2 \cdot 47381 \cdot 1085287719049570327739050925845914539948927360923370110769$
107	$26328072917139301684688220214666205225396172568864115593153438826 =$
	$2 \cdot 862207 \cdot 33602281 \cdot 85871353 \cdot 69807710360281 \cdot 228939975565877 \cdot 331081901714999$
109	$421249166674228800251100330124945140261321879842750041189776992282 =$
	$2 \cdot 2617 \cdot 620764811 \cdot 129651709107106280529021406475320711149271787278988543$
113	$107839786668602557431646595347682461521285605430038087099528386736762 =$
	$2 \cdot 53919893334301278715823297673841230760642802715019043549764193368381$

Table 3: Computing the Cardinality of the Jacobian $\mathbb{J}_{C_0}(\mathbb{F}_{2^n})$

n	$\#\mathbb{J}_{C_0}(\mathbb{F}_{2^n})$
61	$5316911989384839930345585607286135912 =$
	$2^3 \cdot 483853 \cdot 8684228116229 \cdot 158170258164913997$
67	$217780714847749832994997151829687427694 96 =$
	$2^3 \cdot 2722258935596872912437464397871092846187$
71	$55751862997048813677718552804661205240962 48 =$
	$2^3 \cdot 569 \cdot 2699 \cdot 416396257 \cdot 1089801570384585437289692293$
73	$892029807974491853159919527951204824510631 12 =$
	$2^3 \cdot 293 \cdot 263950481 \cdot 5661445943 \cdot 67348577251 \cdot 378132069281$
79	$3653754093325336845142045077054108891232540 89272$
	$2^3 \cdot 798104358750115175106 71 \cdot 572255064965338652342729$
83	$935361047891312674316442962537964128600065 33765592$
	$2^3 \cdot 50242889 \cdot 34520115435043977433 \cdot 674128130756552285 1227$
89	$383123885216451157219690382614340814499889612946264008$
	$2^3 \cdot 179 \cdot 1069 \cdot 83091469 \cdot 301204924452355371151542028498245913 9979$
97	$25108406941546708114295960500655104894931956823678392606472$
	$2^3 \cdot 5825627 \cdot 1755694859485001 \cdot 306858006865407663939079619643509467$
101	$642775217703596425482873021294166049514680686138162640703 5048$
	$2^3 \cdot 19080201689 \cdot 37954942754010986482 5131 \cdot 1109475803736779006300 63959$
103	$1028440348325753718721632039846803428926933523899531557062 45112$
	$2^3 \cdot 4819352903 \cdot 676426898960529275556539 \cdot 394347889652663496781274 5867$
107	$263280729171392916642707936271699702956776360620653167495551 78392$
	$2^3 \cdot 275419 \cdot 1188789908218841 \cdot 2579078640412757953 \cdot 389731486204747038330 5777$
109	$4212491666742286933322438913444243057199045582393547294224085 38088$
	$2^3 \cdot 1338521 \cdot 1375524369017 \cdot 3635750197819 \cdot 3382869865979927 \cdot 2325285384440165921$
113	$1078397866686025609256895253484746322810204769468794551308200 63235464$
	$2^3 \cdot 3617 \cdot 13109 \cdot 123411655021 \cdot 852620315028296881 85249 \cdot 27018367721820145876679009$

5 Improvements

Following the idea of Koblitz [12], we modified our set of possible coefficients and used the set

$$R' = \{0, \pm 1, \pm 2, \pm(1+\tau), \pm(1-\tau), \pm(1-2\tau), \pm 2+\tau\}$$

as the domain of coefficients. Accepting the cost of 6 precomputations and storing these elements (instead of only 2 for set R), this choice enables us to realize a τ-adic expansion in the sense that no two consecutive coefficients are nonzero (cf. [23]). Using u as in the following table we force $a + b\tau + c\tau^2 + d\tau^3 - u$ to be divisible by τ^2, i. e. the next coefficient will be zero. If $4|a$ then $u = 0$, else take

$b \bmod 4 / a \bmod 8$	1	2	3	5	6	7
0	1	2	$-(1-2\tau)$	$1-2\tau$	-2	-1
1	$1+\tau$	$2+\tau$	$-(1+\tau)$	$1-\tau$	$-2+\tau$	$-(1-\tau)$
2	$1-2\tau$	-2	-1	1	2	$-(1-2\tau)$
3	$1-\tau$	$-2+\tau$	$-(1-\tau)$	$1+\tau$	$2+\tau$	$-(1+\tau)$

By using a modified version of Algorithm 3.1, the average density of the expansion was quite lower than 1/2, and the average length was about $2\log_2(m)$ as with the first set. The average length of the reduced τ-adic representations was even $< n + 2$ for an extension of degree n.

In Table 4, we present our experimental results. The generation of the integers m was identical to the one in Table 1. The difference lies in the choice of the set R' and the new τ-adic expansion as described above.

Therefore the expected number of nonzero coefficients c_i is approximately 43.3 percent, and Algorithm 3.5 for computing multiples $m[D]$ of divisor classes needs about $9/20n$ additions of reduced divisors. So with this set R' we need only $\frac{9n}{20}/\frac{5n}{9} = 81$ percent of the operations as with the set R on the cost of more storing and precomputations. Thus, we are able to reduce the costs of the generic double-and-add-method in the Jacobian to approximately $\frac{9n}{20}/3n = 3/20 = 15$ percent.

Table 4: Average Length and Density

n	average length	average density	n	average length	average density
61	63.02	0.4284	97	99.67	0.4177
67	69.00	0.4275	101	102.95	0.4287
71	72.98	0.4288	103	104.93	0.4289
73	32.15	0.4287	107	109.05	0.4288
79	81.01	0.4287	109	111.01	0.4287
83	84.99	0.4286	113	114.96	0.4285
89	91.00	0.4288			

References

[1] Adleman, L., DeMarrais, J., Huang, M.-D.: A Subexponential Algorithm for Discrete Logarithms over the Rational Subgroup of the Jacobians of Large Genus Hyperelliptic Curves over Finite Fields. In: Algorithmic Number Theory Seminar ANTS-I. Lecture Notes in Computer Science, Vol. 877. Springer-Verlag, Berlin Heidelberg New York (1994) 28–40

[2] Cantor, D. G.: Computing in the Jacobian of a Hyperelliptic Cyurve. Mathematics of Computation **48** (1987) 95–101

[3] Diffie, W., Hellman, M. E.: New Directions in Cryptography. IEEE Trans. Inform. Theory **22** (1976) 644–654

[4] ElGamal, T.: A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans. Inf. Theory **IT-31** (1985) 469–472

[5] Frey, G., Rück, H.-G.: A Remark Concerning m-Divisibility and the Discrete Logarithm Problem in the Divisor Class Group of Curves. Mathemathics of Computation **62** (1994) 865–874

[6] Gallant, R., Lambert, R., Vanstone, S.: Improving the Parallelized Pollard Lambda Search on Binary Anomalous Curves. To Appear in Mathematics of Computation. http://www.certicom.com/chal/download/paper.ps

[7] Gaudry, P., Morain, F., Duursma, I.: Speeding Up the Discrete Log Computation on Curves with Automorphisms In: Proc. of The Mathematics of Public Key Cryptography. Fields-Institute Toronto (1999)

[8] Gordon, D.: A Survey of Fast Exponentiation Methods. J. Algs. **27** (1998) 129–146

[9] Koblitz, N.: Elliptic Curve Cryptosystems. Mathemathics of Computation **48** (1987) 203–209

[10] Koblitz, N.: Hyperelliptic Cryptosystems. Journal of Cryptology **1** (1989) 139–150

[11] Koblitz, N.: CM Curves with Good Cryptographic Properties. In: Advances in Cryptology – Crypto '91. Lecture Notes in Computer Science, Vol. 576. Springer-Verlag, Berlin Heidelberg New York (1992) 279–287

[12] Koblitz, N.: An Elliptic Curve Implementation of the Finite Field Digital Signature Algorithm. In: Advances in Cryptology – Crypto '98. Lecture Notes in Computer Science, Vol. 1462. Springer-Verlag, Berlin Heidelberg New York (1998) 327–337

[13] Krieger U.: Anwendung hyperelliptischer Kurven in der Kryptographie. Diploma thesis, Universität Gesamthochschule Essen (1997)

[14] Meier, W., Staffelbach, O.: Efficient Multiplication on Certain Nonsupersingular Elliptic Curves. In: Advances in Cryptology – Crypto '92. Lecture Notes in Computer Science, Vol. 740. Springer-Verlag, Berlin Heidelberg New York (1993) 333–344

[15] Menezes, A., Wu, Y., Zuccherato, R.: An Elementary Introduction to Hyperelliptic Curves. In: Koblitz, N.: Algebraic Aspects of Cryptography. Springer-Verlag, Berlin Heidelberg New York (1998)

[16] Miller, V.: Use of Elliptic Curves in Cryptography. In: Advances in Cryptology – Crypto '85. Lecture Notes in Computer Science, Vol. 218. Springer-Verlag, Berlin Heidelberg New York (1986) 417–426

[17] Müller, V., Stein, A., Thiel, C.: Computing Discrete Logarithms in Real Quadratic Congruence Function Fields of Large Genus. Mathemathics of Computation **68** (1999) 807–822

[18] van Oorschot, P., Wiener, M. J.: Parallel Collision Search with Cryptanalytic Applications. Journal of Cryptology **12** (1999) 1–28

[19] Mumford, D.: Tata Lectures on Theta I, II. Birkhäuser-Verlag, Boston (1983/84)

[20] Paulus, Rück, H.-G.: Real and imaginary quadratic representations of hyperelliptic function fields Logarithms. Mathematics of Computation **68** (1999) 1233–1241

[21] Pollard, J. M.: Kangaroos, Monopoly and Discrete Logarithms. To appear in Journal of Cryptology

[22] Solinas, J.: An Improved Algorithm for Arithmetic on a Family of Elliptic Curves. In: Advances in Cryptology – Crypto '97. Lecture Notes in Computer Science, Vol. 1294. Springer-Verlag, Berlin Heidelberg New York (1997) 357–371

[23] Solinas, J.: Efficient Arithmetic on Koblitz Curves. Techn. Report CORR 99-09, University of Waterloo (1999), 61 pages. http://www.cacr.math.uwaterloo.ca

[24] Spallek, A.: Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosystemen. Ph.D. thesis, Universität Gesamthochschule Essen (1994)

[25] Stein, A.: Sharp Upper Bounds for Arithmetics in Hyperelliptic Function Fields. Techn. Report CORR 99-23, University of Waterloo (1999), 68 pages. Available at http://www.cacr.math.uwaterloo.ca

[26] Stichtenoth, H.: Algebraic Function Fields and Codes. Springer-Verlag, Berlin Heidelberg New York (1993)

[27] Teske, E.: Speeding up Pollard's rho method for computing discrete logarithms. In: Algorithmic Number Theory Seminar ANTS-III. Lecture Notes in Computer Science, Vol. 1423. Springer-Verlag, Berlin Heidelberg New York (1998) 541–554

[28] Wiener, M., Zuccerato, R.: Faster Attacks on Elliptic Curve Cryptosystems. In: Proceedings of SAC, Workshop on Selected Areas in Cryptography. Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg New York (1998). http://grouper.ieee.org/groups/1363/contrib.html