On Redundant τ-Adic Expansions and Non-adjacent Digit Sets

Roberto Maria Avanzi[1,⋆], Clemens Heuberger[2,⋆⋆], and Helmut Prodinger[3,⋆⋆⋆]

[1] Faculty of Mathematics and Horst Görtz Institute for IT Security
Ruhr-University Bochum, Germany
roberto.avanzi AT ruhr-uni-bochum.de
[2] Institut für Mathematik B, Technische Universität Graz, Austria
clemens.heuberger AT tugraz.at
[3] Department of Mathematics, University of Stellenbosch, South Africa
hproding AT sun.ac.za

Abstract. This paper studies τ-adic expansions of scalars, which are important in the design of scalar multiplication algorithms on Koblitz Curves, and are less understood than their binary counterparts.

At Crypto '97 Solinas introduced the width-w τ-adic non-adjacent form for use with Koblitz curves. It is an expansion of integers $z = \sum_{i=0}^{\ell} z_i \tau^i$, where τ is a quadratic integer depending on the curve, such that $z_i \neq 0$ implies $z_{w+i-1} = \ldots = z_{i+1} = 0$, like the sliding window binary recodings of integers. We show that the digit sets described by Solinas, formed by elements of minimal norm in their residue classes, are uniquely determined. However, unlike for binary representations, syntactic constraints do not necessarily imply minimality of weight.

Digit sets that permit recoding of all inputs are characterized, thus extending the line of research begun by Muir and Stinson at SAC 2003 to Koblitz Curves.

Two new useful digit sets are introduced: one set makes precomputations easier, the second set is suitable for low-memory applications, generalising an approach started by Avanzi, Ciet, and Sica at PKC 2004 and continued by several authors since. Results by Solinas, and by Blake, Murty, and Xu are generalized.

Termination, optimality, and cryptographic applications are considered. We show how to perform a "windowed" scalar multiplication on Koblitz curves without doing precomputations first, thus reducing memory storage dependent on the base point to just one point.

⋆ Supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT.

⋆⋆ Supported by the Austrian Science Foundation FWF, project S9606, that is part of the Austrian National Research Network "Analytic Combinatorics and Probabilistic Number Theory."

⋆⋆⋆ Supported by the NRF grant 2053748 of the South African National Research Foundation and by the Center of Experimental Mathematics of the University of Stellenbosch.

E. Biham and A.M. Youssef (Eds.): SAC 2006, LNCS 4356, pp. 285–301, 2007.

1 Introduction

Elliptic curves (EC) [15,17] are now a well established cryptographic primitive. The performance of an EC cryptosystem depends on the efficiency of the fundamental operation, the *scalar multiplication*, i.e. the computation of the multiple sP of a point P by an integer s. Among all EC, *Koblitz curves* [16], defined by the equation

$$E_a\colon y^2 + xy = x^3 + ax^2 + 1 \qquad \text{with} \qquad a \in \{0, 1\} \tag{1}$$

over the finite field $\mathbb{F}_{2^n}$, permit particularly efficient implementation of scalar multiplication. Key to their good performance is the Frobenius endomorphism τ, i.e. the map induced on $E_a(\mathbb{F}_{2^n})$ by the Frobenius automorphism of the field extension $\mathbb{F}_{2^n}/\mathbb{F}_2$, that maps field elements to their squares.

Set $\mu = (-1)^{1-a}$. It is known [24, Section 4.1] that τ permutes the points $P \in E_a(\mathbb{F}_{2^n})$, and $(\tau^2 + 2)P = \mu\tau(P)$. Identify τ with a root of

$$\tau^2 - \mu\tau + 2 = 0 \ . \tag{2}$$

If we write an integer z as $\sum_{i=0}^{\ell} z_i\tau^i$, where the digits z_i belong to a suitably defined digit set $\mathcal{D}$, then we can compute zP as $\sum_{i=0}^{\ell} z_i\tau^i(P)$ via a Horner scheme. The resulting method [16,23,24] is called a "τ-and-add" method since it replaces the doubling with a Frobenius operation in the classic double-and-add scalar multiplication algorithm. Since a Frobenius operation is much faster than a group doubling, scalar multiplication on Koblitz curves is a very fast operation.

The elements dP for all $d \in \mathcal{D}$ are computed before the Horner scheme. Larger digit sets usually correspond to representations $\sum_{i=0}^{\ell} z_i\tau^i$ with fewer non-zero coefficients i.e. to Horner schemes with fewer group additions. Optimal performance is attained upon balancing digit set size and number of non-zero coefficients.

Solinas [23,24] considers the residue classes in $\mathbb{Z}[\tau]$ modulo τ^w which are coprime to τ, and forms a digit set comprising the zero and an element of minimal norm from each residue class coprime to τ. We prove (Theorem 2) that such elements are unique, hence this digit set is uniquely determined. It has cardinality $1 + 2^{w-1}$. Solinas' recoding enjoys the *width-w non-adjacent property*

$$z_i \neq 0 \qquad \text{implies} \qquad z_{w+i-1} = \ldots = z_{i+1} = 0 \ , \tag{3}$$

and is called the τ-adic width-w non-adjacent form (or τ-w-NAF for short). Every integer admits a unique τ-w-NAF.

We call a digit set allowing to recode all integers satisfying property (3) a *(width-w) non-adjacent digit set*, or w-NADS for short. Theorem 1 is a criterion for establishing whether a given digit set is a w-NADS, which is very different in substance from the criterion of Blake, Murty, and Xu [8]. The characterisation of digit sets which allow recoding with a non-adjacency condition is a line of research started by Muir and Stinson in [18] and continued, for example by Heuberger and Prodinger in [11].

Our criterion is applied to digit sets introduced and studied in §§ 2.3 and 2.4. We can prove under which conditions the first set is a w-NADS (Theorem 3), and give precise estimates of the length of the recoding (Theorem 4). The second digit set corresponds, in a suitable sense, to "repeated point halvings" (cf. Theorem 5) and is used to design a width-w scalar multiplication algorithm without precomputations. Among the other results in Section 2 are the facts that the τ-adic w-NAF as defined by Solinas is not optimal, and that it is not possible to compute minimal expansions by a deterministic finite automaton. In Section 3 we discuss the relevance of our results for cryptographic applications and performance. We conclude in Section 4. Due to space constraints, most proofs have been omitted. They will be given in the extended version of the paper.

2 Digit Sets

Let $\mu \in \{\pm 1\}$, τ be a root of equation (2) and $\bar{\tau}$ the complex conjugate of τ. Note that $2/\tau = \bar{\tau} = \mu - \tau = -\mu(1+\tau^2)$. We consider expansions to the base of τ of integers in $\mathbb{Z}[\tau]$. It is well known that $\mathbb{Z}[\tau]$, which is the ring of algebraic integers of $\mathbb{Q}(\sqrt{-7})$, is a unique factorization domain.

Definition 1. *Let $\mathcal{D}$ be a (finite) subset of $\mathbb{Z}[\tau]$ containing 0 and $w \geqslant 1$ be an integer. A* $\mathcal{D}$-expansion *of $z \in \mathbb{Z}[\tau]$ is a sequence $\boldsymbol{\varepsilon} = (\varepsilon_j)_{j \geqslant 0} \in \mathcal{D}^{\mathbb{N}_0}$ such that*

1. *Only a finite number of the digits ε_j is nonzero.*
2. *$\mathsf{value}(\boldsymbol{\varepsilon}) := \sum_{j \geqslant 0} \varepsilon_j \tau^j = z$, i.e., $\boldsymbol{\varepsilon}$ is indeed an expansion of z.*

The Hamming weight *of $\boldsymbol{\varepsilon}$ is the number of nonzero digits ε_j. The* length *of $\boldsymbol{\varepsilon}$ is defined as*

$$\mathsf{length}(\boldsymbol{\varepsilon}) := 1 + \max\{j : \varepsilon_j \neq 0\} \ .$$

*A $\mathcal{D}$-expansion of z is a $\mathcal{D}$-w-*Non-Adjacent-Form ($\mathcal{D}$-w-NAF) of z, *if*

3. *Each block $(\varepsilon_{j+w-1}, \ldots, \varepsilon_j)$ of w consecutive digits contains at most one nonzero digit ε_k, $j \leqslant k \leqslant j+w-1$.*

A $\{0, \pm 1\}$-2-NAF is also called a τ-NAF.

The set $\mathcal{D}$ is called a w-Non-Adjacent-Digit-Set (w-NADS), *if each $z \in \mathbb{Z}[\tau]$ has a $\mathcal{D}$-w-NAF.*

Typically, $\mathcal{D}$ will have cardinality $1 + 2^{w-1}$, but we do not require this in the definition. One of our aims is to investigate which $\mathcal{D}$ are w-NADS, and we shall usually restrict ourselves to digit sets formed by adjoining the 0 to a reduced residue system modulo τ^w, which is defined as usual:

Definition 2. *Let $w \geqslant 1$ a natural number. A reduced residue system $\mathcal{D}'$ for the number ring $\mathbb{Z}[\tau]$ modulo τ^w is a set of representatives for the congruence classes of $\mathbb{Z}[\tau]$ modulo τ^w that are coprime to τ.*

For a digit set $\mathcal{D}$ for $\mathbb{Z}[\tau]$ formed by 0 together with a reduced residue system, Algorithm 1 either recodes an integer $z \in \mathbb{Z}[\tau]$ to the base of τ, or enters in a infinite loop for some inputs when $\mathcal{D}$ is not a NADS.

Algorithm 1. General windowed integer recoding

INPUT: An element z from $\mathbb{Z}[\tau]$, a natural number $w \geqslant 1$ and a reduced residue system $\mathcal{D}'$ for the number ring R modulo τ^w.
OUTPUT: A representation $z = \sum_{j=0}^{\ell-1} z_j \tau^j$ of length ℓ of the integer z with the property that if $z_j \neq 0$ then $z_{j+i} = 0$ for $1 \leqslant i < w$.

1. $j \leftarrow 0, u \leftarrow z$
2. **while** $u \neq 0$ **do**
3. **if** $\tau \mid u$ **then**
4. $z_j \leftarrow 0$ [Output 0]
5. **else**
6. Let $z_j \in \mathcal{D}'$ s.t. $z_j \equiv z \pmod{\tau^w}$ [Output z_j]
7. $u \leftarrow u - z_j,\ u \leftarrow u/\tau,\ j \leftarrow j + 1$
8. $\ell \leftarrow j$
9. **return** $(\{z_j\}_{j=0}^{\ell-1}, \ell)$

Example 1. A digit set obtained by adjoining the zero to a reduced residue system is not necessarily a NADS. This fact has been observed in the binary case in [18]. If we take $w = 1$ and the digit set $\{0, 1 - \tau\}$ (here the reduced residue set modulo $\tau = \tau^1$ comprises the single element $1 - \tau$) we see that the element 1 has an expansion $(1 - \tau) + (1 - \tau)\tau + (1 - \tau)\tau^2 + (1 - \tau)\tau^3 + \cdots$. Algorithm 1 does not terminate in this case.

2.1 Algorithmic Characterization

As already mentioned, one aim of this paper is to investigate which digit sets $\mathcal{D}$ are in fact w-NADS. For concrete $\mathcal{D}$ and w, this question can be decided algorithmically:

Theorem 1. *Let $\mathcal{D}$ be a finite subset of $\mathbb{Z}[\tau]$ containing 0 and $w \geqslant 1$ be an integer. Let*

$$M := \left\lfloor \frac{\max\{N(d) : d \in \mathcal{D}\}}{\left(2^{w/2} - 1\right)^2} \right\rfloor,$$

where $N(z)$ denotes the norm of z, i.e., $N(a + b\tau) = (a + b\tau)(a + b\bar{\tau}) = a^2 + \mu ab + 2b^2$ for $a, b \in \mathbb{Z}$.

Consider the directed graph $G = (V, A)$ defined by its set of vertices

$$V := \{0\} \cup \{z \in \mathbb{Z}[\tau] : N(z) \leqslant M\,, \tau \nmid z\}$$

and set of arcs

$$A := \{(y, z) \in V^2 : \text{There exist } d \in \mathcal{D} \setminus \{0\},\ \text{and } v \geqslant w \text{ s.t. } z = \tau^v y + d\}\ .$$

Then $\mathcal{D}$ is a w-NADS iff the following conditions are both satisfied.

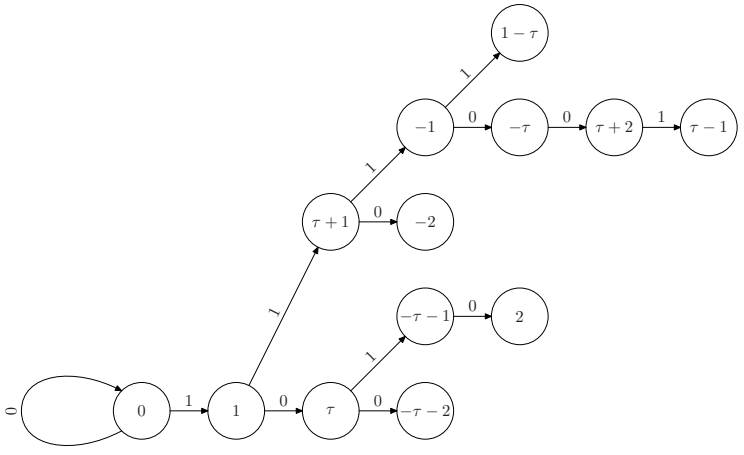

Fig. 1. Directed Graph G for $\mu = -1$, $w = 1$, $\mathcal{D} = \{0, 1\}$. The arcs are labeled with (v, d) as in the definition of the graph, i.e. $y \xrightarrow{(v,d)} z$ means that $z = \tau^v y + d$.

1. *The set $\mathcal{D}$ contains a reduced residue system modulo τ^w.*
2. *In $G = (V, A)$, each vertex $z \in V$ is reachable from 0.*

If $\mathcal{D}$ is a w-NADS and $\mathcal{D} \setminus \{0\}$ is a reduced residue system modulo τ^w, then each $z \in \mathbb{Z}[\tau]$ has a unique $\mathcal{D}$-w-NAF.

We now make some remarks and discuss two well-known examples.

Remark 1. A number $a + \tau b \in \mathbb{Z}[\tau]$ is relatively prime to τ iff a is odd. This follows from the fact that τ is a prime element in $\mathbb{Z}[\tau]$ and that τ divides a rational integer iff the latter is even.

Example 2. Let $w = 1$ and $\mathcal{D} = \{0, 1\}$. By Remark 1, there is only one residue class prime to τ. In this case $M = 5$, so $V = \{0, \pm 1, \pm 1 \pm \tau\}$. The corresponding directed graph in the case $\mu = -1$ is shown in Figure 1. The case $\mu = 1$ is similar.

We see that all 7 states are reachable from 0. Thus, $\{0, 1\}$ is a 1-NADS. This is equivalent to saying that τ is the base of a canonical number system in $\mathbb{Z}[\tau]$ in the sense of [13], and is a particular case of results from [12].

Remark 2. Example 2 implies that there are exactly 2^w residue classes modulo τ^w; a complete residue system is: $\left\{ \sum_{j=0}^{w-1} \varepsilon_j \tau^j \text{ with } \varepsilon_j \in \{0, 1\} \text{ for } 0 \leqslant j < w \right\}$. There are 2^{w-1} residue classes coprime to τ^w, a reduced residue system is: $\left\{ 1 + \sum_{j=1}^{w-1} \varepsilon_j \tau^j \text{ with } \varepsilon_j \in \{0, 1\} \text{ for } 1 \leqslant j < w \right\}$.

Example 3. Let $w = 2$ and $\mathcal{D} = \{0, \pm 1\}$. Using Remark 2, it is easily seen that $\{\pm 1\}$ is a reduced residue system modulo τ^2. In this case, $M = 1$, the graph G consists of the three states $V = \{0, \pm 1\}$ only, and those are obviously reachable from 0. Thus $\{0, \pm 1\}$ is a 2-NADS. This has been proved by Solinas [23,24].

Example 4. Let us consider the digit set $\mathcal{D} = \{0\} \cup \{\pm 1, \pm 3, \ldots, \pm(2^{w-1} - 1)\}$. The odd digits form a reduced residue system modulo τ^w, since τ^w divides a rational integer if and only if 2^w divides it (note that τ and $\bar{\tau}$ are coprime primes in $\mathbb{Z}[\tau]$). However, this digit set is not a w-NADS for all w. For instance, for $w = 6$, the number $1 - \mu\tau$ has no $\mathcal{D}$-6-NAF. Using Theorem 1, we can verify that for $w \in \{2, 3, 4, 5, 7, 8, 9, 10\}$, this set $\mathcal{D}$ is a w-NADS.

2.2 Representatives of Minimal Norm

Theorem 2. *Let τ, $w \geqslant 2$ be as above, and $\mathcal{D}$ a digit set consisting of 0 together with one element of minimal norm from each odd residue class modulo τ^w.*

The digit set $\mathcal{D}$ is uniquely determined. In other words, in each odd residue class modulo τ^w there exists a unique element of minimal norm.

In [5,6] it has been shown that the τ-NAF has minimal weight among all the τ-adic expansions with digit set $\{0, \pm 1\}$. Since the digit set $\mathcal{D} = \{0, \pm 1 \pm \bar{\tau}\}$ is also Solinas' set for $w = 3$, in the same paper it is in fact shown that a $\mathcal{D}$-w-NAF with this digit set is a $\mathcal{D}$-expansion of minimal weight. For the radix 2 the analogous result is known to be true for all positive w [1,19]. So one might conjecture that the same holds for our choice of τ. But, the following example shows that this is not the case:

Example 5. Consider $\mu = -1$, $w = 4$, and the set $\mathcal{D}$ of minimal norm representatives modulo τ^w. We have $\mathcal{D} = \{0, \pm 1, \pm 1 \pm \tau, \pm(3 + \tau)\}$ and

$$\mathsf{value}(1, 0, 0, 0, -1 - \tau, 0, 0, 0, 1 - \tau) = -9 = \mathsf{value}(-3 - \tau, 0, 0, -1) \ .$$

The first expansion is the $\mathcal{D}$-w-NAF and has Hamming weight 3. The second expansion does not satisfy the w-NAF condition, has Hamming weight 2 and is even shorter than the first expansion.

Even worse, we exhibit chaotic behaviour in the following sense: for every integer $k > 0$, a pair of numbers can be found which are congruent modulo τ^k, but whose optimal $\mathcal{D}$-expansions differ even at the least significant position. Thus it is impossible to compute an optimal $\mathcal{D}$-expansion of z by a deterministic transducer automaton or an online algorithm.

Proposition 1. *Let $w = 4$, and $\mathcal{D} = \{0, \pm 1, \pm 1 \pm \tau, \pm(3 - \mu\tau)\}$ (all signs are independent) be the set of minimal norm representatives modulo τ^w. For every nonnegative integer ℓ, we define*

$$\begin{aligned} z_\ell &:= \mathsf{value}\big(\ \ 0, 0, 0, 0, \mu - \tau, (0, 0, 0, -3\mu + \tau)^{(\ell)}, 0, 0, 0, 0, 1 - \mu\tau, 0, 0, 0, -1\big) \,, \\ z'_\ell &:= \mathsf{value}\big(-\mu, 0, 0, 0, \mu - \tau, (0, 0, 0, -3\mu + \tau)^{(\ell)}, 0, 0, 0, 0, 1 - \mu\tau, 0, 0, 0, -1\big) \,, \end{aligned} \tag{4}$$

where $(0, 0, 0, -3\mu + \tau)^{(\ell)}$ means that this four-digit block is repeated ℓ times. Then $z_\ell \equiv z'_\ell \pmod{\tau^{4\ell+13}}$. All $\mathcal{D}$-optimal expansions of z_ℓ are given by

$$\big((0, 0, 0, 3 - \mu\tau)^{(\ell_2)}, 0, 0, \mu - \tau, (0, 0, 0, -3\mu + \tau)^{(\ell_1)}, 0, 0, 0, 0, 1 - \mu\tau, 0, 0, 0, -1\big) \ ,$$

where ℓ_1 and ℓ_2 are nonnegative integers summing up to ℓ. There is only one $\mathcal{D}$-optimal expansion of z'_ℓ, it is given by

$$\left((0,0,0,-3+\mu\tau)^{(\ell+1)},0,0,0,0,-3\mu+\tau,0,0,1+\mu\tau\right) \ .$$

Note that the $\mathcal{D}$-optimal expansion of z'_ℓ has Hamming weight $\ell+3$, whereas the $\mathcal{D}$-w-NAF of z'_ℓ given in (4) has Hamming weight $\ell+4$. The proof is based on the search of shortest paths in an auxiliary automaton.

2.3 Syntactic Sufficient Conditions

The aim of this section is to prove sufficient conditions for families of sets $\mathcal{D}$ to be a w-NADS at the level of digits of the τ-NAF. In contrast to Theorem 1, where a decision can be made for any concrete set $\mathcal{D}$, we will now focus on families of such sets. Blake, Murty, and Xu [8] gave sufficient conditions based on the norm of the numbers involved.

Proposition 2. *Let $w \geqslant 1$ and $\boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon}'$ two τ-NAFs. Then* $\mathsf{value}(\boldsymbol{\varepsilon}) \equiv \mathsf{value}(\boldsymbol{\varepsilon}') \pmod{\tau^w}$ *if and only if*

$$\varepsilon_j = \varepsilon'_j \text{ for } 0 \leqslant j \leqslant w-2 \text{ and } |\varepsilon_{w-1}| = |\varepsilon'_{w-1}| \ . \tag{5}$$

Definition 3. *Let w be a positive integer and $\mathcal{D}$ be a subset of*

$$\{\,0\,\} \cup \{\ \mathsf{value}(\boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon} \text{ is a } \tau\text{-NAF of length at most } w \text{ with } \varepsilon_0 \neq 0\ \}$$

consisting of 0 *and a reduced residue system modulo τ^w. Then $\mathcal{D}$ is called a* set of short τ-NAF representatives *for τ^w.*

By Proposition 2, an example for a set of short τ-NAF representatives is

$$\mathcal{D} = \{\,0\,\} \cup \Big\{\ \mathsf{value}(\boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon} \text{ is a } \tau\text{-NAF of length at most } w \text{ with } \varepsilon_0 \neq 0 \text{ and } \varepsilon_{w-1} \in \{0,\varepsilon_0\}\ \Big\} \ . \tag{6}$$

All other sets of short τ-NAF representatives are obtained by changing the signs of ε_{w-1} without changing ε_0 in some of the $\boldsymbol{\varepsilon}$. It is easy to check that the cardinality of $\mathcal{D}$ is indeed $1+2^{w-1}$.

The main result of this section is the following theorem, which states that in almost all cases, a set of short τ-NAF representatives is a w-NADS:

Theorem 3. *Let w be a positive integer and $\mathcal{D}$ a set of short τ-NAF representatives. Then $\mathcal{D}$ is a w-NADS iff it is not in the following table*

w	μ	$\mathcal{D}$	*Remark*
3	-1	$\{1,-1,-\tau^2+1,-\tau^2-1\}$	$(-\tau-1)\left(1-\tau^3\right) = -\tau^2+1$
3	-1	$\{1,-1,-\tau^2+1,\ \ \tau^2-1\}$	$(-\tau-1)\left(1-\tau^3\right) = -\tau^2+1$
3	-1	$\{1,-1,\ \ \tau^2+1,\ \ \tau^2-1\}$	$(\tau+1)\left(1-\tau^3\right) = \tau^2-1$
3	1	$\{1,-1,-\tau^2+1,\ \ \tau^2-1\}$	$(-\tau+1)\left(1-\tau^6\right) = \left(-\tau^2+1\right)\tau^3+\tau^2-1$

(the "Remark" column contains an example of an element which cannot be represented). In particular, if $w \geqslant 4$, then $\mathcal{D}$ is always a w-NADS.

The following result is concerned with the lengths of recodings that make use of the set of short τ-NAF representatives.

Theorem 4. *Let $w \geqslant 2$ be a positive integer, $\mathcal{D}$ a set of short τ-NAF representatives, and $\boldsymbol{\varepsilon}$ a $\mathcal{D}$-w-NAF of some $z \in \mathbb{Z}[\tau]$.*

Then the length of $\boldsymbol{\varepsilon}$ can be bounded by

$$2\log_2|z| - w - 0.18829 < \mathsf{length}(\boldsymbol{\varepsilon}) < 2\log_2|z| + 7.08685\ , \qquad \text{if } w \geqslant 4\ , \tag{7}$$

$$2\log_2|z| - 2.61267 < \mathsf{length}(\boldsymbol{\varepsilon}) < 2\log_2|z| + 5.01498\ , \qquad \text{if } w = 3\ , \tag{8}$$

$$2\log_2|z| - 0.54627 < \mathsf{length}(\boldsymbol{\varepsilon}) < 2\log_2|z| + 3.51559\ , \qquad \text{if } w = 2\ . \tag{9}$$

Note that (9) is Solinas' [24] Equation (53).

2.4 Point Halving

For any given point P, point halving [14,22] consists in computing a point Q such that $2Q = P$. This operation applies to all elliptic curves over binary fields. Its evaluation is (at least two times) faster than that of a doubling and a halve-and-add scalar multiplication algorithm based on halving instead of doubling can be devised. This method is not useful for Koblitz curves because halving is slower than a Frobenius operation.

In [3] it is proposed to insert a halving in the "τ-and-add" method to speed up Koblitz curve scalar multiplication. This approach brings a non-negligible speedup and was refined in [5,6], where the insertion of a halving was interpreted as a digit set extension as follows: Inserting a halving in the scalar multiplication is equivalent to adding $\pm\bar{\tau}$ to the digit set $\{0, \pm 1\}$. Note that, by Theorem 3, $\mathcal{D} = \{0, \pm 1, \pm\bar{\tau}\}$ is the only 3-NADS of short τ-NAF representatives. In particular $\mathcal{D}' = \{\pm 1, \pm\bar{\tau}\}$ is a reduced residue system modulo τ^3.

The next two theorems state that more powers of $\bar{\tau}$ still produce reduced residue systems $\mathcal{D}'$, which in some cases give rise to w-NADS.

Theorem 5. *Let $w \geqslant 2$. Then $\mathcal{D}' := \{\pm\bar{\tau}^k : 0 \leqslant k < 2^{w-2}\}$ is a reduced residue system modulo τ^w.*

Theorem 6. *Let $\mathcal{D} := \{0\} \cup \{\pm\bar{\tau}^k : 0 \leqslant k < 2^{w-2}\}$. If $w \in \{2, 3, 4, 5, 6\}$ then $\mathcal{D}$ is a w-NADS. If $w \in \{7, 8, 9, 10, 11, 12\}$ then $\mathcal{D}$ is not a w-NADS.*

Sketch of the Proof of Theorem 6. For every pair (w, μ) with $w \leqslant 6$ the conditions of Theorem 1 have been verified by heavy symbolic computations.

For $7 \leqslant w \leqslant 12$ and both values of μ the graph G contains loops that are not reachable from 0. In other words, there are elements in $\mathbb{Z}[\tau]$ that have periodic expansions. For example, if $w = 7$ and $\mu = 1$ we have

$$-9 + 34\,\tau = \frac{\bar{\tau}^{27} - \bar{\tau}^6\tau^7}{1 - \tau^{16}} = \bar{\tau}^{27} - \bar{\tau}^6\tau^7 + \bar{\tau}^{27}\tau^{16} - \bar{\tau}^6\tau^{23} + \bar{\tau}^{27}\tau^{32} - \cdots$$

and the number $371 - 20\,\tau$ has for all w with $8 \leqslant w \leqslant 12$ (also with $\mu = 1$) periodic expansion $(\bar{\tau}^{41} - \bar{\tau}^5\tau^{12})(1 - \tau^{24})^{-1}$. □

2.5 Comparing the Digit Sets

So far, three digit sets have been studied: the minimal norm representatives, short NAF representatives, and powers of $\bar{\tau}$. It is a natural question to ask what are the relations between these sets when they are w-NADS.

The minimal norm representatives are exactly the powers of $\bar{\tau}$ for $w \leqslant 4$. For the same range of w, all digits of these digit sets have a τ-NAF of length at most w, which implies that they are also digit sets of short NAF representatives.

If symmetry is required, i.e., if d is a digit, then $-d$ must also be a digit, by Theorem 3 there is only *one* w-NADS of short NAF representatives for $w \leqslant 3$: it coincides with the digit sets of minimal norm representatives and powers of $\bar{\tau}$. For $w = 4$, however, there is a symmetric w-NADS of short NAF representatives distinct from the other two digit sets. For $w \geqslant 5$, the three concepts are different: the lengths of the τ-NAFs of the powers of $\bar{\tau}$ grow exponentially in w, and the lengths of some minimal norm representatives exceed w slightly (at most by 2).

The table below summarizes the above considerations and provides further information. "MNR" stands for the minimal norm representatives digit set, whereas "P$\bar{\tau}$" stands for the powers of $\bar{\tau}$. The last two rows show the maximum length of the τ-NAFs of the digits.

w	2	3	4	5	6
MNR=P$\bar{\tau}$	True	True	True	False	False
Max τ-NAF length MNR	1	3	4	6	8
Max τ-NAF length P$\bar{\tau}$	1	3	4	8	17

3 Applications

All digit sets seen so far can be used in a τ-and-add scalar multiplication, where we first precompute dP for all $d \in \mathcal{D} \setminus \{0\}$ and then we evaluate the scheme $\sum z_i \tau^i(P)$; in fact, only a half of the precomputations usually suffice since in all cases that we explicitly described the non-zero elements of the digit set come in pairs of elements of opposite sign.

The digit set from § 2.3 simplifies the precomputation phase. The digit set from § 2.4 allows us to perform precomputations very quickly or to get rid of them completely. In the next two subsections we shall consider these facts in detail. In § 3.3 we explain how to use digit sets which are not w-NADS when they contain a subset that is a k-NADS for smaller k.

3.1 Using the Short-NAF Digit Set

Let us consider here the digit set $\mathcal{D}$ defined in (6). With respect to Solinas' set it has the advantage of being syntactically defined. If a computer has to work with different curves, different scalar sizes and thus with different optimal choices for the window size, the representatives in Solinas' set must be recomputed – or they must be retrieved from a set of tables. In some cases, the time to compute representatives of minimal norm may have to be subsumed in the total scalar

multiplication time. This is not the case with our set. This flexibility is also particularly important for computer algebra systems.

The scalar is first recoded as a τ-NAF, and the elements of $\mathcal{D}$ are associated to NAFs of length at most w with non-vanishing least significant digit, and thus to certain *odd integers* in the interval $[-a_w, a_w]$ where $a_w = \frac{2^{w+1}-2(-1)^w}{3} - 1$. These integers can be used to index the elements in the precomputation table. We need only to precompute the multiples of the base point by "positive" short NAFs (i.e. with most significant digit equal to 1) – and the corresponding integers are the odd integers in the interval $[0, a_{w-1}]$ together with the integers $\equiv 1 \pmod 4$ in $[a_{w-1}+2, a_w]$. The indices in the table are then obtained by easy compression. The precomputed elements for the scalar multiplication loop can thus be retrieved upon direct reading the τ-NAF, of which we need only to compute the least w significant places. If the least and the w-th least significant digits of this segment of the τ-NAF are both non-zero and have different signs, a carry is generated: Thus, the computation of the τ-NAF should be interleaved with the parsing for short NAFs.

3.2 τ-Adic Scalar Multiplication with Repeated Halvings

Let $w \geqslant 2$ be an integer and $\mathcal{D}$ the digit set defined in § 2.4. Let P be a point on an elliptic curve and $Q_j := \tau^j(2^{-j}P)$ for $0 \leqslant j < 2^{w-2}$ and $R := Q_{2^{w-2}-1}$. To compute zP, we have to compute yR for $y := \bar{\tau}^{2^{w-2}-1}z$. Computing a $\mathcal{D}$-w-NAF of y, this can be done by using the points Q_j, $0 \leqslant j < 2^{w-2}$ as precomputations.

Now, a point halving on an elliptic curve is much faster than a point doubling, and a point addition is not faster than a doubling. Now, with, say, Solinas' set or the short τ-NAF representatives the precomputations always involve at least one addition per digit set element. With our set we require a halving per digit set element. Hence, our approach with the points Q_j and halvings is already faster than traditional ones.

But we can do even better, especially if normal bases are used to represent the field $\mathbb{F}_{2^n}$. Algorithm 2 computes zP using an expansion $y = \sum_{i=0}^{\ell} y_i\tau^i$ of the integer $y := \bar{\tau}^{2^{w-2}-1}z$ where the digits y_i belong to the digit set introduced in Theorem 5, i.e. $\mathcal{D} := \{0\} \cup \{\pm\bar{\tau}^k : 0 \leqslant k < 2^{w-2}\}$.

To explain how it works we introduce some notation. Write $y_i = \varepsilon_i\bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$. We also define

$$y^{(k)} = \sum_{i\,:\,0\leqslant i\leqslant \ell,\ y_i=\pm\bar{\tau}^k} \varepsilon_i\tau^i \quad .$$

Now $y = \sum_{k=0}^{2^{w-2}-1} y^{(k)}\bar{\tau}^k$ and therefore

$$\begin{aligned} zP = \bar{\tau}^{-(2^{w-2}-1)}yP &= \left(\sum_{m=0}^{2^{w-2}-1} y^{(m)}\bar{\tau}^m\right)\bar{\tau}^{-(2^{w-2}-1)}P \\ &= \sum_{m=0}^{2^{w-2}-1} y^{(m)}\bar{\tau}^{m-(2^{w-2}-1)}P = \sum_{m=0}^{2^{w-2}-1} \left(\frac{\tau}{2}\right)^{2^{w-2}-1-m}(y^{(m)})P \quad . \end{aligned}$$

Algorithm 2. τ-adic Scalar Multiplication with Repeated Halvings

INPUT: A Koblitz curve E_a, a point P of odd order on it, and a scalar z.
OUTPUT: zP

1. $y \leftarrow \bar{\tau}^{2^{w-2}-1} z$
   Write $y = \sum_{i=0}^{\ell} y_i \tau^i$ where $y_i \in \mathcal{D} := \{0\} \cup \pm\{\bar{\tau}^k : 0 \leqslant k < 2^{w-2}\}$
   Write $y_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$
2. $\ell_k \leftarrow \max\big(\{-1\} \cup \{i : y_i = \pm\bar{\tau}^k \text{ for some } k\}\big)$
3. $X \leftarrow 0$
4. **for** $k = 0$ **to** $2^{w-2} - 1$ **do**
5. **if** $k > 0$ **then** $X \leftarrow \tau^{n-\ell_k} X$, $X \leftarrow \frac{1}{2} X$
6. **for** $i = \ell_k$ **to** 0 **do**
7. $X \leftarrow \tau X$
8. **if** $y_i = \pm\bar{\tau}^k$ **then** $X \leftarrow X + \varepsilon_i P$
9. **return** (X)

The last expression is evaluated by a Horner scheme in $\frac{\tau}{2}$, i.e. by repeated applications of τ and a point halving, interleaved with additions of $y^{(0)}P$, $y^{(1)}P$, etc. The elements $y^{(k)}P$ are computed by a τ-and-add loop as usual. To save a memory register, instead of computing $y^{(k)}P$ and then adding it to a partial evaluation of the Horner scheme, we apply τ to the negative of the length of $y^{(k)}$ (which is $1 + \ell_k$) to the intermediate result X and perform the τ-and-add loop to evaluate $y^{(k)}P$ starting with this X instead of a "clean" zero. In Step 5 there is an optimization already present in [3]: n is added to the exponent (since $n \approx \ell_k$ and τ^n acts like the identity on the curve) and the operation is also partially fused to the subsequent $\frac{\tau}{2}$. At the end of the internal loop the relation $X = \sum_{m=0}^{k} \left(\frac{\tau}{2}\right)^{k-m} y^{(m)}P$ holds, thus proving the correctness.

Apart from the input, we only need to store the additional variable X and the recoding of the scalar. The multiplication of z by $\bar{\tau}^{2^{w-2}-1}$ is an easy operation, and the negative powers of τ can be easily eliminated by multiplying by a suitable power of τ^n which operates trivially on the points of the curve. Reduction of this scalar by $(\tau^n - 1)/(\tau - 1)$ following [23,24] is also necessary.

An issue with Algorithm 2 is that the number of Frobenius operations may increase exponentially with w, since the internal loop is repeated up to 2^{w-2} times. This is not a problem if a normal basis is used to represent the field, but may induce a performance penalty with a polynomial basis. A similar problem was faced by the authors of [20], and they solved it adapting an idea from [21]. The idea consists in keeping a copy R of the point P in normal basis representation. Instead of computing $y^{(k)}P$ by a Horner scheme in τ, the summands $\varepsilon_i \tau^i P$ are just added together. The power of the Frobenius is applied to R *before* converting the result back to a polynomial basis representation and accumulating it. According to [10] a basis conversion takes about the same time as one polynomial basis multiplication, and the two conversion routines require each a matrix that occupies $O(n^2)$ bits of memory.

Algorithm 3. Low-memory τ-adic Scalar Multiplication on Koblitz Curves with Repeated Halvings, for Fast Inversion

INPUT: $P \in E(\mathbb{F}_{2^n})$, scalar z
OUTPUT: zP

1. $y \leftarrow \bar{\tau}^{2^{w-2}-1} z$
   Write $y = \sum_{i=0}^{\ell} y_i \tau^i$ where $y_i \in \mathcal{D} := \{0\} \cup \pm\{\bar{\tau}^k : 0 \leqslant k < 2^{w-2}\}$
   Write $y_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$
2. $R \leftarrow \mathrm{normal\_basis}(P)$
3. $Q \leftarrow 0$
4. **for** $k = 0$ **to** $2^{w-2} - 1$
5. **if** $k > 0$ **then** $Q \leftarrow \tau Q$, $Q \leftarrow \frac{1}{2} Q$
6. **for** $i = 0$ **to** ℓ
7. **if** $y_i = \pm \bar{\tau}^k$ **then** $Q \leftarrow Q + \varepsilon_i \mathrm{polynomial\_basis}(\tau^i R)$
8. **return** Q

Algorithm 4. Low-memory τ-adic Scalar Multiplication on Koblitz Curves with Repeated Doublings, for Slow Inversion

INPUT: $P \in E(\mathbb{F}_{2^n})$, scalar z
OUTPUT: zP

1. Write $z = \sum_{i=0}^{\ell} z_i \tau^i$ where $z_i \in \mathcal{D} := \{0\} \cup \pm\{\bar{\tau}^k : 0 \leqslant k < 2^{w-2}\}$
   Write $z_i = \varepsilon_i \bar{\tau}^{k_i}$ with $\varepsilon_i \in \{0, \pm 1\}$
2. $R \leftarrow \mathrm{normal\_basis}(P)$ [Keep in affine coordinates]
3. $Q \leftarrow 0$ [Q is in Lopez-Dahab coodinates]
4. **for** $k = 2^{w-2} - 1$ **to** 0
5. **if** $k > 0$ **then** $Q \leftarrow \tau^{-1} Q$, $Q \leftarrow 2Q$ [τ^{-1} is three square roots]
6. **for** $i = 0$ **to** ℓ
7. **if** $z_i = \pm \bar{\tau}^k$ **then** $Q \leftarrow Q + \varepsilon_i \mathrm{polynomial\_basis}(\tau^i R)$ [Mixed coord.]
8. **return** Q [Convert to affine coordinates]

Algorithm 3 is a realisation of this approach. It is suited in the context where a polynomial basis is used for a field and the cost of an inversion is not prohibitive. The routines normal_basis and polynomial_basis convert the coordinates of the points between polynomial and normal bases.

Algorithm 4 is the version for fields with a slow inversion (such as large fields). It uses inversion-free coordinate systems and, since no halving formula is known for such coordinates, a doubling is used instead of a halving. This is not a problem, since using Projective or López-Dahab coordinates (see [9, § 15.1]) a doubling followed by an application of τ^{-1} (which amounts to three square root extractions) is about twice as fast as a mixed-coordinate addition preceded by a basis conversion, hence the situation is advantageous as the previous one. This also dispenses us with the need of using an auxiliary scalar y.

Algorithm 5. Windowed Integer Recoding With Termination Guarantee

INPUT: An element z from $\mathbb{Z}[\tau]$, a natural number $w \geqslant 1$ and a set of reduced residue systems $\mathcal{D}'_k \subset \mathcal{D}'_{k+1} \subset \ldots \subset \mathcal{D}'_w$ modulo $\tau^k, \tau^{k+1}, \ldots, \tau^w$ respectively, $(1 \leqslant k < w)$ where $\mathcal{D}'_k \cup \{0\}$ is a k-NADS.
OUTPUT: A representation $z = \sum_{j=0}^{\ell-1} z_j \tau^j$ of length ℓ.

1. $j \leftarrow 0, u \leftarrow z, v \leftarrow w$
2. **while** $u \neq 0$ **do**
3. **if** $\tau \mid u$ **then**
4. $z_j \leftarrow 0$
5. **else**
6. Let $z_j \in \mathcal{D}'_v$ s.t. $z_j \equiv u \pmod{\tau^v}$
7. **if** $(|z_j| \geqslant |u|(2^{v/2} - 1)$ AND $v > k)$ **then** decrease v and retry:
8. $v \leftarrow v - 1$, go to Step 6
9. $u \leftarrow u - z_j,\ u \leftarrow u/\tau,\ j \leftarrow j + 1$
10. $\ell \leftarrow j$
11. **return** $(\{z_j\}_{j=0}^{\ell-1}, \ell)$

Although the digit set $\mathcal{D}$ introduced in Theorem 5 is not a w-NADS for all w, in the next subsection we show how to save the situation.

3.3 Stepping Down Window Size

Let $\mathcal{D}_w$ be a family of digit sets, parametrized by an integer w, which are w-NADS for some small values of w, but not in general, and such that $\mathcal{D}_{v-1} \subset \mathcal{D}_v$ for all v. Then, Algorithm 1 may enter a loop for a few inputs. This can be caused by the appearance of "large" digits towards the end of the main loop of the recoding algorithm. Then the norm of the variable u gets too small in comparison to the chosen digit, and $|u| \leqslant \left|\frac{u-z_j}{\tau^w}\right| \leqslant \frac{|u|+|z_j|}{2^{w/2}}$. For most other inputs the algorithm terminated and delivers the expected low density. How can we save it? One possibility is to decrease w for the rest of the computation, so that the corresponding digit set is a NADS. We call this operation *stepping down*. The resulting recoding may have a slightly higher weight, but the algorithm is guaranteed to terminate. One possible implementation is presented as Algorithm 5.

Solinas can prove that his τ-adic w-NAF terminates because his digits are representants of minimal norm, and have norm bounded by $\frac{4}{7}2^w$. The presence of digits of non-minimal norm is a necessary but not sufficient condition for non-termination. In fact, the digit sets from Example 4 and from § 2.4 are w-NADS with some digits of norm larger than 2^w.

Remark 3. The digit set from Example 4, the syntactically defined set of § 2.3 and the set of Theorem 5 all have the property that each set is contained in the sets with larger w – hence Algorithm 5 can be used.

Remark 4. Checking an absolute value (or a norm) in Algorithm 5, Step 7 is expensive. Hence we need an alternative strategy. Let M_w be defined as M in Theorem 1 for the digit set we are considering, with parameter w. Consider an easy function that is bounded by the norm: for example, if $z = a + b\tau$, $\lambda(z) = \max\{\lceil |a + \frac{\mu}{2}b| \rceil^2, 2\lceil |\frac{\mu}{4}a + b| \rceil^2\}$. It is easy to check that $\lambda(z) \leqslant N(z)$ and that $\lambda(z) = 0$ iff $z = 0$. Therefore, if $\lceil \log_2(M_v) \rceil \geqslant \lfloor \log_2(\lambda(z)) \rfloor$ we step down to a new value of v with $\lceil \log_2(M_v) \rceil < \lfloor \log_2(\lambda(z)) \rfloor$. These checks are quickly computed only by using the bit lengths of a and b and performing additions, subtractions and bit shifts (but no expensive multiplication). The values $\lceil \log_2(M_v) \rceil$ are precomputed in an easy way.

Remark 5. In our experiments, the recodings done with the different digit sets have similar length, the average density is $1/(w+1)$ (see also § 3.4), and stepping down only marginally increases the weight. Thus, the new digit sets bring their advantages with *de facto* no performance penalty.

3.4 A Performance Remark

Algorithms 2, 3 and 4 compute scalar multiplications by performing $2^{w-2} - 1$ "faster" operation blocks and (roughly) $n/(w+1)$ "slower" operation blocks. In Algorithm 2 (with normal bases) the two block types are given by a halving, resp. an addition. In Algorithm 3 (resp. 4) these two block types are given by a Frobenius operation plus a halving (resp. by an inverse Frobenius plus a doubling), and by a basis conversion followed by an addition (for both algorithms). In all cases we can see that computing the first block takes α times the time for the second block, where $\alpha \leqslant 1/2$.

We now determine asymptotically optimal values for w in these algorithms in terms of n, where n is assumed to be large. This will lead to large values w, such that the digit set from § 2.4 is probably not a w-NADS. We will therefore have to use Algorithm 5 (or a variant of it). For the sake of simplicity, we do not decrease v step by step depending on the norm of $|z_j|$, but we use $v = w$ for $j < L$ and $v = 6$ for $j \geqslant L$, where the parameter L will be chosen below.

Let z be a random integer in $\mathbb{Z}[\tau]$ with $|z| \leqslant |\tau|^n$. Here "random" means that for every positive integer m, every residue class modulo τ^m is equally likely. Let $y = \sum_{j=0}^{L-1} z_j \tau^j$ where the z_j are calculated by Algorithm 5. Then $y \equiv z \pmod{\tau^L}$ and $|y| \leqslant |\tau|^{2^{w-2}-1+L-1}(1 + |\tau|^{-w})^{-1}$. Thus $|(z-y)/\tau^L| \leqslant |\tau|^{n-L} + |\tau|^{2^{w-2}-2}$. The choice

$$L = n - 2^{w-2} + 2$$

implies that $|(z-y)/\tau^L| \leqslant 2|\tau|^{n-L}$. The expected length of the $\mathcal{D}_6$-6-NAF of $(z-y)/\tau^L$ is $n - L + O(1)$. Here, $\mathcal{D}_6 = \{0\} \cup \{\pm\bar{\tau}^k : 0 \leqslant k < 16\}$. We conclude that the expected Hamming weight of the expansion returned by Algorithm 5 is

$$\frac{L}{w+1} + \frac{n-L}{7} + O(1) \ .$$

Here, we use the well-known fact that a v-NAF of length m has expected Hamming weight $m/(v+1) + O(1)$.

Algorithm 2 performs $2^{w-2}-1$ point halvings, the number of additions being given by the Hamming weight of the expansion. With α as above, the total costs of the curve operations (measured in additions) is

$$\alpha 2^{w-2} + \frac{L}{w+1} + \frac{n-L}{7} + O(1) = 2^{w-2}\left(\alpha + \frac{1}{7}\right) + \frac{n-2^{w-2}}{w+1} + O(1) \ .$$

Balancing the two main terms gives

$$\hat{w} = \frac{1}{\log 2} W\left(\frac{7 \cdot 2^{\frac{21\alpha+10}{7\alpha+1}} \log 2}{7\alpha+1} n\right) - \frac{7\alpha+8}{7\alpha+1} \ .$$

where W is the main branch of Lambert's W function. Asymptotically, this is $\hat{w} = \log_2 n - \log_2 \log_2 n + 2 - \log_2\left(\alpha + \frac{1}{7}\right) + O\left(\frac{\log\log n}{\log n}\right)$. Thus we choose

$$w = \left\lfloor \log_2 n - \log_2 \log_2 n + 2 - \log_2\left(\alpha + \frac{1}{7}\right) \right\rfloor$$

and see that the expected number of curve additions asymptotically equals

$$\frac{n}{\log_2 n}\left(1 + c + O\left(\frac{\log\log n}{\log n}\right)\right) \tag{10}$$

with $\frac{1}{2} < c = 2^{-\{\log_2 n - \log_2 \log_2 n + 2 - \log_2\left(\alpha+\frac{1}{7}\right)\}} \leqslant 1$.

For Algorithms 3 and 4, the unit in the cost (10) is given by the cost of a group addition and a base conversion – the latter being comparable to a field multiplication. We thus have the following result:

Theorem 7. *Algorithms 2, 3 and 4 are sublinear scalar multiplication algorithms on Koblitz Curves with constant input-dependent memory consumption.*

Note that here *sublinear* refers to the number of group operations, and "constant memory consumption" refers to the number of registers required for temporary variables – each one taking of course $O(n)$ bits. Usual windowed methods with precomputations have, of course, similar time complexity but use storage for $2^{w-2}-1$ points [23,24] and thus $O(n2^w) = O(n^2/\log n)$ bits of memory. Algorithms 3 and 4 need $O(n^2)$ bits of field-dependent (but not point-dependent) data for base conversion (as in [21,20]) that can be stored statically (such as in ROM).

For the same values of w, our algorithms perform better than techniques storing precomputations. The precomputation stage with Solinas' digit set takes one addition and some Frobenius operations per precomputation. Using the digit set from § 2.4 these additions can be replaced with cheaper operations (halvings or doublings depending on the coordinate system), whereas in Algorithms 3 and 4 the cost of the basis conversion associated to each addition in the main loop is relatively small. In all cases, the increase in recoding weight is marginal. A more precise performance evaluation (including small values of n and w) lies beyond the scope of this paper; however, in [2] some simple operation counts and comparisons with other methods can be found. The method in [7] is also sublinear, but its applicability still has to be assessed – the authors warn that the involved constants may be quite large. See [4] for another approach.

4 Conclusions

The paper at hand presents several new results about τ-adic recodings.

Digit sets allowing a w-NAF to be computed for all inputs are characterised. We study digit sets with interesting properties for Koblitz curves.

We prove that Solinas' digit set, characterised by the property that the elements have minimal norm, is uniquely determined. We show, by means of an example, that the non adjacency property does not imply minimality of weight, and enunciate a result implying that optimal expansions cannot be computed by a deterministic finite automaton.

In § 2.3 we introduce a new digit set characterised by syntactic properties. Its usage is described in § 3.1.

The digit set introduced in § 2.4 together with Algorithms 2, 3 and 4 permit to perform a "windowed" τ-adic scalar multiplication without requiring storage for precomputed points. This is potentially useful for implementation on restricted devices. Our methods can perform better than previous methods that make use of precomputations. Some operation counts (based on the performance of real-world implementations of finite field arithmetic) comparing our algorithms with other methods can be found in [2]. Better performance assessments are part of future work.

Acknowledgements. This paper was partly written during a joint visit of the first two authors to the Department of Mathematical Sciences, Stellenbosch University, and during a visit (supported by the FWF project S9606) of the first author to the Institut für Mathematik B, Technische Universität Graz. The authors thank these institutions for their hospitality.

References

1. Avanzi, R.M.: A Note on the Signed Sliding Window Integer Recoding and its Left-to-Right Analogue. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 130–143. Springer, Heidelberg (2004)
2. Avanzi, R.M.: Delaying and Merging Operations in Scalar Multiplication: Applications to Curve-Based Cryptosystems. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS. vol. 4356, pp. 203–219, Springer, Heidelberg
3. Avanzi, R.M., Ciet, M., Sica, F.: Faster Scalar Multiplication on Koblitz Curves combining Point Halving with the Frobenius Endomorphism. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 28–40. Springer, Heidelberg (2004)
4. Avanzi, R.M., Dimitrov, V., Doche, C., Sica, F.: Extending Scalar Multiplication using Double Bases. In: ASIACRYPT 2006, LNCS. vol. 4284, pp. 130–144, Springer, Heidelberg (2006)

5. Avanzi, R.M., Heuberger, C., Prodinger, H.: Minimality of the Hamming Weight of the τ-NAF for Koblitz Curves and Improved Combination with Point Halving. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 332–344. Springer, Heidelberg (2006)
6. Avanzi, R.M., Heuberger, C., Prodinger, H.: Scalar Multiplication on Koblitz Curves Using the Frobenius Endomorphism and its Combination with Point Halving: Extensions and Mathematical Analysis. Algorithmica 46, 249–270 (2006)
7. Avanzi, R.M., Sica, F.: Scalar Multiplication on Koblitz Curves Using Double Bases. Cryptology ePrint Archive. In: VIETCRYPT 2006, LNCS. vol. 4341, pp. 131–146, Springer, Heidelberg (2006)
8. Blake, I.F., Murty, V.K., Xu, G.: A note on window τ-NAF algorithm. Information Processing Letters 95, 496–502 (2005)
9. Cohen, H., Frey, G. (eds.): The Handbook of Elliptic and Hyperelliptic Curve Cryptography. CRC Press, Boca Raton (2005)
10. Coron, J.-S., M'Raïhi, D., Tymen, C.: Fast generation of pairs $(k, [k]p)$ for Koblitz elliptic curves. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 151–164. Springer, Heidelberg (2001)
11. Heuberger, C., Prodinger, H.: Analysis of Alternative Digit Sets for Nonadjacent Representations. Monatshefte für Mathematik, pp. 219–248 (2006)
12. Kátai, I., Kovács, B.: Canonical number systems in imaginary quadratic fields. Acta Math. Hungar. 37, 159–164 (1981)
13. Kátai, I., Szabó, J.: Canonical Number Systems for Complex Integers. Acta Scientiarum Mathematicarum 1975, 255–260
14. Knudsen, E.W.: Elliptic Scalar Multiplication Using Point Halving. In: Lam, K.-Y., Okamoto, E., Xing, C. (eds.) ASIACRYPT 1999. LNCS, vol. 1716, pp. 135–149. Springer, Heidelberg (1999)
15. Koblitz, N.: Elliptic curve cryptosystems. Math. Comp. 48, 203–209 (1987)
16. Koblitz, N.: CM-curves with good cryptographic properties. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 279–287. Springer, Heidelberg (1992)
17. 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)
18. Muir, J.A., Stinson, D.R.: Alternative digit sets for nonadjacent representations. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 306–319. Springer, Heidelberg (2004)
19. Muir, J.A., Stinson, D.R.: Minimality and other properties of the width-w nonadjacent form. Math. Comp. 75, 369–384 (2006)
20. Okeya, K., Takagi, T., Vuillaume, C.: Short Memory Scalar Multiplication on Koblitz Curves. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 91–105. Springer, Heidelberg (2005)
21. Park, D.J., Sim, S.G., Lee, P.J.: Fast scalar multiplication method using change-of-basis matrix to prevent power analysis attacks on Koblitz curves. In: Chae, K.-J., Yung, M. (eds.) Information Security Applications. LNCS, vol. 2908, pp. 474–488. Springer, Heidelberg (2004)
22. Schroeppel, R.: Elliptic curve point ambiguity resolution apparatus and method. International Application Number PCT/US00/31014 (filed 9 November, 2000)
23. Solinas, J.A.: An improved algorithm for arithmetic on a family of elliptic curves. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 357–371. Springer, Heidelberg (1997)
24. Solinas, J.A.: Efficient Arithmetic on Koblitz Curves. Designs, Codes and Cryptography 19(2/3), 125–179 (2000)