A New Minimal Average Weight Representation for Left-to-Right Point Multiplication Methods

M. Khabbazian and T.A. Gulliver
Department of Electrical & Computer Engineering
University of Victoria
majidk@ece.ubc.ca

August 2, 2004

Abstract

This paper introduces a new radix-2 representation with the same average weight as the width-w nonadjacent form (w-NAF). In both w-NAF and the proposed representations, each nonzero digit is an odd integer with absolute value less than M. However, for w-NAF, M is of the form 2^{w-1}, while for the proposed representation it can be any positive integer. Therefore, using the proposed integer representation we can use the available memory efficiently, which is attractive for devices with limited memory. Another advantage of the proposed representation over w-NAF is that it can be obtained by scanning the bits from left-to-right. This property is also useful for memory-constrained devices because it can reduce both time and space complexity of fast point multiplication techniques.

1 Introduction

In 1985 Koblitz [1] and Miller [2] proposed the group of points on an elliptic curve over a finite field to create the elliptic curve discrete logarithm problem (ECDLP). Unlike the discrete logarithm problem over the multiplicative group of a finite field (DLP), there is no known subexponential-time algorithm to solve the ECDLP in general. Thus, elliptic curves are an attractive alternative to implement many cryptographic techniques such as Diffie-Hellman [3] and ElGamal [4].

Point multiplication dominates the execution time of elliptic curve cryptosystems, so various methods have been studied to enhance the performance of this operation [5, 6]. Specifically, the integer representation of the multiplier plays an important role in the performance of these methods [7]. Among the existing integer representations, those with minimal average Hamming weight (number of nonzero digits) such as w-NAF are more attractive. This is due to the fact that they reduce the required number of point additions/subtractions. It is also of interest to have a representation which can be obtained by scanning the bits from left to right (i.e., from the most significant bit to the least significant bit) [8, 9]. This property eliminates the need for recoding and storing the multiplier in advance, hence improving the performance of left-to-right point multiplication methods in terms of running time and memory space. We are also interested in representations which allow efficient usage of memory. This is important for devices with a small amount of memory. In [10], the author proposes a generalized w-NAF which allows more efficient usage of memory than w-NAF. However, similar to the w-NAF recoding algorithm, the generalized w-NAF recoding algorithm scans the bits from right to left.

In this paper, we introduce a new integer representation which has the same average weight as w-NAF, but has the advantage that it results in a point multiplication method which uses less memory. This memory saving results from the fact that the new representation can be obtained by scanning the multiplier bits from left to right. It can also result in more efficient usage of memory. This is because in the left-to-right point multiplication algorithms (such as Algorithm 1, given in Section 2.2), we need to store m points where m is the number of positive integers in the digit set. For w-NAF and the representations proposed in [11, 12], m is of the form 2^{w-1}, where w is a positive integer. So if we have for example sufficient memory for storing seven points in a device with small amount of storage, using these representations we can only store up to four points and the rest of the memory is wasted. However, using the proposed representation we are able to use all available memory to store seven points. This is due to the fact that for the proposed representation m can be any positive integer.

The rest of this paper is organized as follows. In Section 2, we briefly review elliptic curves. Section 3 proposes the new integer representation and presents an efficient recoding algorithm to generate it. In Section 4, we analyze the average Hamming weight of the proposed representation. In Section 5, we show how the proposed representation can be used for computing elliptic curve point multiplication. Finally, some conclusions are given in section 6.

2 Elliptic Curves

An elliptic curve E over the field $\mathbb{F}$ is a smooth curve in the so called "long weirestraß form"

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

where $a_1, \ldots, a_6 \in \mathbb{F}$. Equation 1 may be simplified to

$$E : y^2 = x^3 + ax + b \tag{2}$$

$$E : y^2 + xy = x^3 + ax^2 + b \tag{3}$$

When $Char(\mathbb{F}) \neq 2, 3$ and $Char(\mathbb{F}) = 2$, respectively.

2.1 Point Addition

Let $E(\mathbb{F})$ be the set of points $P = (x, y) \in \mathbb{F}^2$ that satisfy the elliptic curve equation (along with a "point at infinity" denoted $\mathcal{O}$). It is well known that $E(\mathbb{F})$ together with the point addition operation given in Table 1 form an abelian group. As shown in Table 2, point inversion in this group can be computed easily. Note that in both Tables 1 and 2, the points $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ and $P_3 = (x_3, y_3)$ are represented in affine coordinates.

$Char(\mathbb{F}) = 2$	$P_1 \neq \pm P_2$	$\lambda = \frac{y_2+y_1}{x_2+x_1}$ $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$ $y_3 = (x_1 + x_3)\lambda + y_1 + x_3$
$Char(\mathbb{F}) = 2$	$P_1 = P_2$	$\lambda = \frac{y_1}{x_1} + x_1$ $x_3 = \lambda^2 + \lambda + a$ $y_3 = (x_1 + x_3)\lambda + y_1 + x_3$
$Char(\mathbb{F}) \neq 2, 3$	$P_1 \neq \pm P_2$	$\lambda = \frac{y_2-y_1}{x_2-x_1}$ $x_3 = \lambda^2 - x_1 - x_2$ $y_3 = (x_1 - x_3)\lambda - y_1$
$Char(\mathbb{F}) \neq 2, 3$	$P_1 = P_2$	$\lambda = \frac{3x_1^2+a}{2y_1}$ $x_3 = \lambda^2 - 2x_1$ $y_3 = (x_1 - x_3)\lambda - y_1$

Table 1: Elliptic curve point addition ($P_3 = P_1 + P_2$).

$Char(\mathbb{F}) = 2$	$-P_1 = (x_1, x_1 + y_1)$
$Char(\mathbb{F}) \neq 2, 3$	$-P_1 = (x_1, -y_1)$

Table 2: Elliptic curve point inversion.

2.2 Point Multiplication

Elliptic curve cryptographic techniques require the computation of point multiplication defined as repeated addition

$$kP = \underbrace{P + P + \ldots + P}_{k \text{ times}},$$

where P is an elliptic curve point, $k \in [1, n-1]$ is an integer, and n is the order of the point P. To compute the point multiplication kP, the multiplier k can be represented in base 2 as

$$k = \sum_{0 \leq i < l} k_i 2^i,$$

where $k_i \in B \cup \{0\}$ and B is a set of nonzero integers including 1. Then, Algorithm 1 can use this representation of k to compute kP. Note that in the precomputation stage of this algorithm, we are only required to compute and store dP for positive integers $d \in B$. This is because point inversion in $E(\mathbb{F})$ requires almost no computational effort. In the evaluation stage of this algorithm, we require about l point doublings (it is in fact slightly less) and $(H(k) - 1)$ point additions, where $H(k)$ denotes the Hamming weight of the representation of k.

3 The Proposed Representation

Let $B = \{\pm 1, \pm 3, \ldots, \pm(2^{w-1} - 1)\}$ and $k = \sum_{0 \leq i < l} b_i 2^i$, $b_i \in \{0, 1\}$ be an integer. Then k can be represented as $k = \sum_{0 \leq i \leq l} k_i 2^i$, $k_i \in B \cup \{0\}$ such that of any w consecutive k_i's at most one is nonzero. This is a unique representation of k called w-NAF. It is known that w-NAF has the minimum $H(k)$ among all radix-2 representations of k with digits in B [13]. This property makes w-NAF suitable for efficient implementation of point multiplication. However, the w-NAF recoding algorithm is a right-to-left algorithm, so we need to store the w-NAF representation of the multiplier before using it in a left-to-right point multiplication algorithm (such as Algorithm 1). In contrast to w-NAF, the proposed representation can be obtained using a left-to-right algorithm (Algorithm 2). In addition, in the proposed representation,

---

Algorithm 1: Left-to-right point multiplication

Input: An integer $k = \sum_{0 \le i < l} k_i 2^i$, $k_i \in B \cup \{0\}$, $P \in E(\mathbb{F})$.
Output: kP.

Precomputation Stage:
1. Compute and store dP for all positive integers $d \in B$.

Evaluation Stage:
2. $Q \leftarrow \mathcal{O}$.
3. for i from $l-1$ down to 0 do
3.1 $Q \leftarrow 2Q$.
3.2 if $k_i > 0$ then $Q \leftarrow Q + k_i P$.
3.3 else if $k_i < 0$ then $Q \leftarrow Q - (-k_i)P$.
4. return Q.

---

we are able to use a more general set $B = \{\pm 1, \pm 3, \ldots, \pm(2m-1)\}$, where m is equal to the number of points we want to store in the precomputation stage. This feature allows us to use the available memory efficiently.

For example, assume that we have enough memory to store three points. As a result, we can use the set $B = \{\pm 1, \pm 3, \pm 5\}$. Let

$$k = (1011011101010111111001011001000 1)_2.$$

Using Algorithm 2, k is recoded to

$$k = (\underbrace{1\bar{1}1}_{3}\,0\,\underbrace{\bar{1}1}_{-1}\,00\,\underbrace{\bar{1}1\bar{1}1}_{-5}\,\underbrace{\bar{1}1}_{-1}\,00000\,\underbrace{\bar{1}01}_{-3}\,\underbrace{\bar{1}10\bar{1}}_{-5}\,0\,\underbrace{1\bar{1}}_{1}\,00\,\underbrace{1\bar{1}}_{1})_2,$$

where $\bar{1}$ denotes -1.

4 Analysis of the Proposed Representation

In this section we analyze the average Hamming weight of the proposed representation. To compare the Hamming weight of the proposed representation with that of w-NAF, we use the same set B as used for w-NAF, namely $B = \{\pm 1, \pm 3, \ldots, \pm(2^{w-1}-1)\}$. Let

$$f(n) = \frac{1}{2^{n-1}} \sum_{2^{n-1} \le N < 2^n} H(N), \tag{4}$$

and

$$g(n) = \frac{1}{2^n} \sum_{0 \le N < 2^n} H(N). \tag{5}$$

Algorithm 2: The proposed recoding algorithm

Input: Integers m and $k = \sum_{0 \le i < l} b_i 2^i$, $b_i \in \{0, 1\}$.
Output: A sequence of pairs $\{(k_i, e_i)\}_{i=0}^{d-1}$ such that $0 \le e_i \le l$, $k_i \in \{\pm 1, \pm 3, \ldots, \pm(2m-1)\}$, and $k = \sum_{0 \le i < d} k_i 2^{e_i}$.

Suppose that $b_l = b_{-1} = 0$ and let $\bar{b}_i = b_{i-1} - b_i$ for $0 \le i \le l$, so we have $\bar{b}_i \in \{0, 1, -1\}$ and $(\bar{b}_l, \bar{b}_{l-1}, \ldots, \bar{b}_0)_2 = (b_{l-1}, b_{l-2}, \ldots, b_0)_2$.

1. $i \leftarrow l$, $j \leftarrow 0$.
2. while $i \ge 0$ do
2.1 if $\bar{b}_i = 0$ then $i \leftarrow i - 1$.
2.2 else
2.2.1 Let t be the minimum integer such that $|(\bar{b}_i, \bar{b}_{i-1}, \ldots \bar{b}_t)_2|$ is an odd integer less than $2m$.
2.2.2 $k_j \leftarrow (\bar{b}_i, \bar{b}_{i-1}, \ldots \bar{b}_t)_2$, $e_j \leftarrow t$, $i \leftarrow t - 1$, $j \leftarrow j + 1$.

$f(n)$ is the average Hamming weight of all n-bit numbers while $g(n)$ is the average Hamming weight of all m-bit numbers where $m \le n$. From (4) and (5) we have

$$\begin{aligned} 2^n g(n) &= \sum_{0 \le N < 2^n} H(N) = \sum_{0 \le N < 2^{n-1}} H(N) + \sum_{2^{n-1} \le N < 2^n} H(N) \\ &= 2^{n-1} g(n-1) + 2^{n-1} f(n). \end{aligned}$$

Therefore, the two functions are related through the equation

$$f(n) = 2g(n) - g(n-1). \tag{6}$$

Now suppose that $2^{n+w-2} \le N < 2^{n+w-1}$ is an integer. Therefore, N can be represented as

$$N = \sum_{0 \le i < n+w-1} b_i 2^i, \quad b_i \in \{0, 1\}$$

and

$$N = \sum_{0 \le i < n+w} \bar{b}_i 2^i, \quad \bar{b}_i \in \{0, 1, -1\}, \bar{b}_i = b_{i-1} - b_i.$$

These are the binary representation and the signed binary representation of N, respectively (note that to obtain the signed binary representation we assume that $b_{n+w-1} = b_{-1} = 0$).We can also write N uniquely as $N = q2^n + r$ with $2^{w-2} \leq q < 2^{w-1}$ and $0 \leq r < 2^n$.

We will consider two cases, first when $b_{n-1} = 0$ and then when $b_{n-1} = 1$. For Case 1 we have

$$(\bar{b}_{n+w-1}, \bar{b}_{n+w-2}, \ldots, \bar{b}_n)_2 = (b_{n+w-2}, b_{n+w-3}, \ldots b_n)_2 = q,$$

and

$$(\bar{b}_{n-1}, \bar{b}_{n-2}, \ldots, \bar{b}_0)_2 = (b_{n-1}, b_{n-2}, \ldots b_0)_2 = r.$$

Since $q \leq 2^{w-1} - 1$,we deduce that $H(N) = 1 + H(r)$.
For Case 2 we have

$$(\bar{b}_{n+w-1}, \bar{b}_{n+w-2}, \ldots, \bar{b}_n)_2 = (b_{n+w-2}, b_{n+w-3}, \ldots b_n)_2 + 1 = q + 1$$

and

$$(\bar{b}_{n-1}, \bar{b}_{n-2}, \ldots, \bar{b}_0)_2 = (b_{n-1}, b_{n-2}, \ldots b_0)_2 - 2^n = r - 2^n.$$

Therefore, $H(N) = 1 + H(r - 2^n) = 1 + H(2^n - r)$ (note that $H(M) = H(-M)$). These two cases can be summarized to

$$H(N) = \begin{cases} 1 + H(r) & \text{if } 0 \leq r < 2^{n-1} \\ 1 + H(2^n - r) & \text{if } 2^{n-1} \leq r < 2^n. \end{cases} \tag{7}$$

Then it follows by writing $N = q2^n + r$ that

$$\begin{aligned} 2^{n+w-2} f(n+w-1) &= \sum_{2^{n+w-2} \leq N < 2^{n+w-1}} H(N) \\ &= 2^{w-2} \sum_{0 \leq r < 2^{n-1}} (1 + H(r)) + 2^{w-2} \sum_{2^{n-1} \leq r < 2^n} (1 + H(2^n - r)) \\ &= 2^{w-2} \sum_{0 \leq r < 2^{n-1}} (1 + H(r)) + 2^{w-2} \sum_{0 < r \leq 2^{n-1}} (1 + H(r)) \\ &= 2^{w-2}(2^{n-1} g(n-1) + 2^{n-1}) + 2^{w-2}(2^{n-1} g(n-1) + 2^{n-1} + 1) \\ &= 2^{n+w-2} g(n-1) + 2^{n+w-2} + 2^{w-2}, \end{aligned}$$

so that

$$f(n+w-1) = g(n-1) + 1 + 2^{-n}. \tag{8}$$

Combining (8) with (6) we obtain the recursion

$$2g(n+w) - g(n+w-1) - g(n) = 1 + 2^{-(n+1)},$$

with the initial conditions

$$g(n) = \begin{cases} 1 - 2^{-n} & \text{if } 1 \le n \le w - 1 \\ \frac{5}{4} - 2^{-w} & \text{if } n = w. \end{cases}$$

Now let $u(n) = g(n) - \frac{n}{w+1} + 2^{-(n+1)}$. It is easy to see that $u(n)$ satisfies the homogeneous recursion

$$2u(n+w) - u(n+w-1) - u(n) = 0,$$

with the initial conditions

$$u(n) = \begin{cases} 1 - \frac{n}{w+1} - 2^{-(n+1)} & \text{if } 1 \le n \le w - 1 \\ \frac{1}{4} + \frac{1}{w+1} - 2^{-(w+1)} & \text{if } n = w. \end{cases}$$

To solve this recursion we use the following proposition from [14].

Proposition 1. *Let $u(n)$ be a sequence satisfying the linear recursion $2u(n+w) - u(n+w-1) - u(n) = 0$ for $n \ge 1$. Then there exists $\rho > 1$ such that as $n \to \infty$*

$$u(n) = \frac{1}{w+1}\left(2u(w) + \sum_{1 \le i < w} u(i)\right) + O(\rho^{-n}).$$

Therefore, when $n \to \infty$ we can write

$$\begin{aligned} u(n) &= \frac{1}{w+1}\left(2(\frac{1}{4} + \frac{1}{w+1} - 2^{-(w+1)}) + \sum_{1 \le i < w} u(i)\right) + O(\rho^{-n}) \\ &= \frac{1}{w+1}\left((\frac{1}{2} + \frac{2}{w+1} - 2^{-w}) + ((w-1) - \frac{w(w-1)}{2(w+1)} - (\frac{1}{2} - 2^{-w}))\right) + O(\rho^{-n}) \\ &= \frac{1}{w+1}\left(\frac{1}{w+1} + \frac{w}{2}\right) + O(\rho^{-n}). \end{aligned}$$

Hence

$$g(n) = u(n) + \frac{n}{w+1} - 2^{-(n+1)} = \frac{n}{w+1} + \frac{1}{w+1}(\frac{1}{w+1} + \frac{w}{2}) + O(\rho^{-n}) - 2^{-(n+1)}$$

Applying (8) gives

$$\begin{aligned} f(n) &= g(n-w) + 1 + 2^{-(n-w+1)} \\ &= \frac{n-w}{w+1} + \frac{1}{w+1}(\frac{1}{w+1} + \frac{w}{2}) + 1 - 2^{-(n-w+1)} + 2^{-(n-w+1)} + O(\rho^{-(n-w)}) \\ &= \frac{n}{w+1} - \frac{(w-1)(w+2)}{2(w+1)^2} + 1 + O(\rho'^{-n}), \rho' > 1. \end{aligned}$$

This is equal to the average Hamming weight of w-NAF when $n \to \infty$ [14]. (Note that in [14], the author compute the average number of group multiplications, which is equal to the average Hamming weight of w-NAF minus 1).

5 A Point Multiplication Algorithm Using the Proposed Representation

As shown in Section 3, the proposed representation can be obtained by scanning the bits from left to right. Consequently, to compute kP, the multiplier doesn't need to be recoded and stored in advance. In other words, recoding k and computing kP can be carried out simultaneously (Algorithm 3). Note that we can use a direct $2^r P$ computation algorithm [15, 16] to compute $2^{i-t+1}Q$ in line 3.3 of Algorithm 3 to further improve the speed of the algorithm.

Algorithm 3: Computing kP using the proposed representation

Input: An integer $k = \sum_{0 \leq i < l} b_i 2^i$, $b_i \in \{0, 1\}$.
Output: kP.

Suppose that $b_l = b_{-1} = 0$ and let $\bar{b_i} = b_{i-1} - b_i$ for $0 \leq i \leq l$.

Precomputation Stage:

1. Compute and store dP for all positive odd integers $d < 2m$, where m is the number of points we want to store in the precomputation stage.

Evaluation Stage:

2. $Q \leftarrow \mathcal{O}$, $i \leftarrow l$.
3. while $i \geq 0$ do
3.1 Let t be the minimum integer such that $|(\bar{b}_i, \bar{b}_{i-1}, \ldots \bar{b}_t)_2|$ is zero or an odd integer less than $2m$.
3.2 $h \leftarrow (\bar{b}_i, \bar{b}_{i-1}, \ldots \bar{b}_t)_2$.
3.3 $Q \leftarrow 2^{i-t+1}Q$, $i \leftarrow t - 1$.
3.4 if $h > 0$ then $Q \leftarrow Q + hP$.
3.5 else if $h < 0$ then $Q \leftarrow Q - (-h)P$.
4. return Q.

---

6 Conclusion

In this paper, a new radix-2 representation was proposed. The average Hamming weight of the proposed representation was analyzed, and shown to be equal to the average Hamming weight of w-NAF. This is an important property, as it speeds up point multiplication methods by reducing the required number of point additions/subtractions.

The proposed representation has two other properties, which are especially attractive for efficient implementation of point multiplication methods in devices with a small amount of memory. First, the proposed representation can be obtained by scanning the bits from left to right. Consequently, there is no need to store the recoded representation in advance, and hence the memory requirements are reduced. Second, using the proposed representation, we can store as many points as we want (up to the limits of available memory). Therefore, we can use memory efficiently to improve the performance of the multiplication methods.

References

[1] N. Koblitz, "Elliptic curve cryptosysmtems," *Math. Comp.*, vol. 48, pp. 203–209, 1987.

[2] V. Miller, "Uses of elliptic curves in cryptography," *Springer-Verlag Lecture Notes in Computer Science*, vol. 218, pp. 417–426, 1987.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. 22, pp. 644–654, 1976.

[4] T. ElGamal, "A public key cryptosystem and a signature scheme based on the discrete logarithm," *IEEE Trans. Inform. Theory*, vol. 31, pp. 469–472, 1985.

[5] I. Blake, G. Seroussy, and N. Smart, *Elliptic Curves in Cryptography.* Cambridge: Cambridge University Press, 1999.

[6] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography.* CRC Press, 1997.

[7] D. Gordon, "A survay of fast exponentiation methods," *J. Algorithms*, vol. 27, pp. 129–146, 1998.

[8] V. Müller, "Fast multiplication on elliptic curves over small fields of characteristic two," *J. Crypt.*, vol. 11, pp. 219–234, 1998.

[9] J. A. Solinas, "Efficient arithmetic on Koblitz curves," *Designs, Codes and Cryptography*, vol. 19, pp. 195–249, 2000.

[10] B. Möller, "Improved techniques for fast exponentiation," *Springer-Verlag Lecture Notes in Computer Science*, vol. 2587, pp. 298–312, 2003.

[11] M. Joye and S. M. Yen, "Optimal left-to-right binary signed-digit recoding," *IEEE Trans. Comput.*, vol. 49, pp. 740–748, 2000.

[12] J. A. Muir and D. R. Stinson, "New minimal weight representations for left-to-right window methods," *Preprint*, 2004, Available from http://www.cacr.math.uwaterloo.ca/tech_reports.html.

[13] J. A. Muir and D. R. Stinson, "Minimality and other properties of the with-w nonadjacent form," *Preprint*, 2004, Available from http://www.cacr.math.uwaterloo.ca/tech_reports.html.

[14] H. Cohen, "Analysis of the flexible window powering algorithm," *Preprint*, 2004, Available from http://www.math.u-bordeaux.fr/~cohen/window.dvi.

[15] J. Guajardo and C. Paar, "Efficient algorithms for elliptic curve cryptosystems," *Springer-Verlag Lecture Notes in Computer Science*, vol. 1294, pp. 342–356, 1997.

[16] Y. Sakai and K. Sakurai, "Efficient scalar multiplications on elliptic curves with direct computations of several doublings," *IEICE Trans. Fundamentals*, vol. E84-A, pp. 120–129, 2001.