Digit Set Randomization in Elliptic Curve Cryptography

David Jao[1], S. Ramesh Raju[2,3], and Ramarathnam Venkatesan[3,4]

[1] University of Waterloo, Waterloo ON N2L3G1, Canada
djao@math.uwaterloo.ca
[2] Theoretical Computer Science Lab, IIT Madras, Chennai – 600036, India
srraju@cse.iitm.ernet.in
[3] Microsoft Research India Private Limited, "Scientia", No:196/36,
2nd Main Road, Sadashivnagar, Bangalore – 560080, India
[4] Microsoft Research, 1 Microsoft Way, Redmond WA 98052, USA
venkie@microsoft.com

Abstract. We introduce a new approach for randomizing the digit sets of binary integer representations used in elliptic curve cryptography, and present a formal analysis of the sparsity of such representations. The motivation is to improve the sparseness of integer representations and to provide a tool for defense against side channel attacks. Existing alternative digit sets D such as $D = \{0, 1, -1\}$ require a certain non-adjacency property (no two successive digits are non-zero) in order to attain the desired level of sparseness. Our digit sets do not rely on the non-adjacency property, which in any case is only possible for a certain very restricted class of digit sets, but nevertheless achieve better sparsity. For example, we construct a large explicit family of digit sets for which the resulting integer representations consist on average of 74% zeros, which is an improvement over the 67% sparsity available using non-adjacent form representations. Our proof of the sparsity result is novel and is dramatically simpler than the existing analyses of non-adjacent form representations available in the literature, in addition to being more general. We conclude with some performance comparisons and an analysis of the resilience of our implementation against side channel attacks under an attack model called the *open representation model*. We emphasize that our side channel analysis remains preliminary and that our attack model represents only a first step in devising a formal framework for assessing the security of randomized representations as a side channel attack countermeasure.

Key words: randomized representations, elliptic curve cryptography, non-adjacent form representations, side channel attack countermeasures.

1 Introduction

Let α be an elliptic curve private key. In traditional elliptic curve cryptography, a point of the form αQ is computed via repeated doubling and addition using the binary representation $\alpha = a_k a_{k-1} \dots a_1 a_0$ of α. By exploiting the fact that inverses on an elliptic curve are easy to compute, one can speed up the computation of αQ using signed binary representations [2, 10]. As a simple example, consider the case where the integers a_i are taken from the set $\{0, 1, -1\}$. In this case, the resulting representations $\alpha = \sum_{i=0}^{k} a_i 2^i$ are no longer unique, but Reitweisner [22] observed in 1960 that these representations become unique if one decrees that no two consecutive a_i are nonzero. The resulting representations are known as *non-adjacent form* representations or NAF representations in the literature. Furthermore, the NAF representation of α is guaranteed to have the fewest possible nonzero terms out of all possible representations of α using $\{0, 1, -1\}$, a property which is desirable for performance reasons because nonzero terms slow down the computation of αQ. Morain and Olivos [14] were among the first to exploit $\{0, 1, -1\}$-representations to speed up elliptic curve computations.

Recently, Muir and Stinson [15] studied representations of the form $\alpha = \sum_{i=0}^{k} a_i$, where $a_i \in \{0, 1, x\}$ for some constant x, and found an infinite (but exponentially rare) class of sets $\{0, 1, x\}$, called *non-adjacent digit sets* or NADS, satisfying the property that each integer α has a unique NAF representation in $\{0, 1, x\}$. Subsequent work [1, 7] has extended the understanding of the properties of NADS and their corresponding NAF representations, but such research has had at best limited applicability to cryptography because of the exponential rarity of known NADS, which results in only a limited variety of such sets being available for use in implementations.

In this paper we introduce and study binary representations with respect to more general digit sets of the form $\{0, 1, x, y, \ldots z\}$. We show that the high performance characteristics of traditional signed binary representations can be realized over this much larger and more general collection of digit sets. Our result enables an entire new class of algorithms for runtime randomization of elliptic curve exponentiation, based on randomized digit sets. We provide both theoretical and empirical analysis showing that EC exponentiation using randomized sparse representations is superior to traditional exponentiation or signed exponentiation in efficiency. Our theoretical analysis is simpler than prior investigations even when restricted to the special case of non-adjacent digit sets of the form $\{0, 1, x\}$, but our results also apply more generally to digit sets having size $2^c + 1$ for any c, with only mild restrictions (e.g. the set must contain one element congruent to $3 \bmod 4$). We achieve this ease of analysis by allowing the use of integer representations which occasionally violate the nonadjacency rule. Nevertheless, we show that these representations have zero density asymptotically equal to or better than the uniquely defined representations arising from NADS. Finally, we provide an analysis indicating that the information entropy of an integer multiplier is lower bounded by that of the digit set under an attack model which we call the *open representation model*, in which the symbolic representation of the integer multiplier (that is, the pattern of digits appearing in the representation) is exposed to the attacker via side channel information but the digit set itself is hidden. The use of randomized digit sets is crucial to this analysis, because otherwise there is no distinction between knowing the symbolic representation of an integer and knowing the integer itself. Since all known side channel attacks to date (for example, [9, 11, 19]) operate by obtaining the symbolic representation of an integer, we believe that the introduction of randomized digit sets and the creation of such a distinction under the open representation model constitutes a crucial first step in devising a rigorous framework for analyzing side channel attack countermeasures. We emphasize, however, that our preliminary investigations fall short of a complete framework for side channel attack analysis and that much more remains to be done in this area.

2 Statistical Properties of NAF Representations

Heuberger and Prodinger [7] recently showed that non-adjacent form representations with respect to digit sets $\{0, 1, x\}$ have an average density of nonzero terms equal to $1/3$, using a detailed combinatorial study involving recurrences. In this section we give a Markov Chain analysis for the $\{0, 1, x\}$ case, which as we will see generalizes readily to larger digit sets. We begin with the relevant definitions.

Definition 2.1. *A* digit set *is a finite set of integers containing both* 0 *and* 1 *as elements.*

For the rest of this section, we assume the digit set D has the form $D = \{0, 1, x\}$ where $x \equiv 3 \pmod 4$ is negative.

Definition 2.2. *Let D be a digit set and let α be a nonnegative integer. A* non-adjacent form *representation (or* NAF *representation) of α with respect to D is a finite (possibly empty) sequence of integers $a_i \in D, i = 0, \ldots, k$, with $a_k \neq 0$, such that $\alpha = \sum_{i=0}^{k} a_i 2^i$, and no two consecutive values of a_i are both nonzero.*

We note at this point that an arbitrary integer α does not necessarily have a NAF representation with respect to D. For the moment, we will limit our attention to the case where α does have a NAF representation with respect to D. Later we will discuss how to modify our algorithm and analysis to apply to the cases where it does not.

Theorem 2.3 ([15]) *Every nonnegative integer has at most one* NAF *representation with respect to D.*

We write $\alpha = (a_k \cdots a_2 a_1 a_0)_2$ to denote that the sequence a_i is the NAF representation for α. By convention, the empty sequence is the NAF representation for 0.

The following definition and theorem provide a method for computing NAF representations.

Definition 2.4. *For any digit set $D = \{0, 1, x\}$, let $f_D \colon \mathbb{N} \to \mathbb{N}$ and $g_D \colon \mathbb{N} \to D \cup (D \times D)$ be the functions defined by*

$$f_D(n) = \begin{cases} n/4 & n \equiv 0 \pmod 4 \\ (n-1)/4 & n \equiv 1 \pmod 4 \\ n/2 & n \equiv 2 \pmod 4 \\ (n-x)/4 & n \equiv 3 \pmod 4 \end{cases}, \qquad g_D(n) = \begin{cases} (0,0) & n \equiv 0 \pmod 4 \\ (0,1) & n \equiv 1 \pmod 4 \\ 0 & n \equiv 2 \pmod 4 \\ (0,x) & n \equiv 3 \pmod 4 \end{cases}.$$

Theorem 2.5 ([15]) *A nonnegative integer α has a* NAF *representation if and only if $f_D(\alpha)$ has a* NAF *representation. Moreover, if $\alpha = (a_k \cdots a_2 a_1 a_0)_2$ and $f_D(\alpha) = (b_\ell \cdots b_2 b_1 b_0)_2$, then $a_k \cdots a_2 a_1 a_0 = b_\ell \cdots b_2 b_1 b_0 \;||\; g_D(\alpha)$, where $||$ denotes concatenation of sequences.*

Theorem 2.5 suggests the following algorithm $\mathcal{A}$ for computing the NAF representation of α:

1. If $\alpha = 0$, then return the empty string. Otherwise:
2. Evaluate $f_D(\alpha)$ and $g_D(\alpha)$.
3. Recursively call the algorithm $\mathcal{A}$ on the new input value $f_D(\alpha)$ in order to find the NAF representation of $f_D(\alpha)$.
4. Concatenate the NAF representation of $f_D(\alpha)$ with $g_D(\alpha)$, and remove any leading zeros, in order to obtain the NAF representation for α.

By Theorems 2.3 and 2.5, the algorithm $\mathcal{A}$ is guaranteed to return the NAF representation of α whenever α has one.

2.1 $\mathcal{A}$ As a Dynamical System

The execution profile of the algorithm $\mathcal{A}$ involves calculating the quantities $\alpha_1 = f_D(\alpha)$, $\alpha_2 = f_D(\alpha_1) = f_D^2(\alpha)$, $\alpha_3 = f_D(\alpha_2) = f_D^3(\alpha)$, etc., as well as the values of $g_D(\alpha)$, $g_D(\alpha_1)$, $g_D(\alpha_2)$, etc. We are interested in knowing the distribution of the integers $\alpha_k \bmod 4$ in order to predict which of the execution pathways for f_D and g_D in Definition 2.4 are more likely to be encountered.

Theorem 2.6 *For a fixed digit set $D = \{0, 1, x\}$, where $x \equiv 3 \pmod 4$ the probability distribution of the congruence classes $\alpha_k \bmod 4$ over the values $(0, 1, 2, 3)$, for random uniformly selected integers $\alpha \in [0, N]$, where $N \gg |x|$, converges to the vector $(\frac{1}{5}, \frac{3}{10}, \frac{1}{5}, \frac{3}{10})$ as $k \to \infty$, with error bounded in magnitude by an exponential in k.*

Proof. By hypothesis, the initial (uniformly selected) input value α has probability distribution $\mathcal{P}_0 = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ over the congruence classes mod 4. The probability distribution $\mathcal{P}_1$ for α_1 is computed as follows:

- By assumption, α is uniformly distributed in $[0, N]$.
- If $\alpha \equiv 0 \pmod 4$, then $\alpha_1 = \alpha/4$ is uniformly distributed mod 4.
- If $\alpha \equiv 1 \pmod 4$, then $\alpha_1 = \frac{\alpha-1}{4}$ is uniformly distributed mod 4.
- If $\alpha \equiv 2 \pmod 4$, then $\alpha_1 = \alpha/2$ is uniformly either 1 or 3 mod 4.
- If $\alpha \equiv 3 \pmod 4$, then $\alpha_1 = \frac{\alpha-x}{4}$ is uniformly distributed mod 4.

Denote by A the matrix

$$A = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}.$$

Then the probability distribution $\mathcal{P}_1$ of α_1 is given by

$$\mathcal{P}_1 = A \cdot \mathcal{P}_0 = \left(\tfrac{3}{16}, \tfrac{5}{16}, \tfrac{3}{16}, \tfrac{5}{16}\right). \tag{2.1}$$

Similarly, the probability distribution $\mathcal{P}_2$ of α_2 is given by

$$\mathcal{P}_2 = A \cdot \mathcal{P}_1 = \left(\tfrac{13}{64}, \tfrac{19}{64}, \tfrac{13}{64}, \tfrac{19}{64}\right). \tag{2.2}$$

In general, the probability distribution $\mathcal{P}_k$ of α_k is given by the formula $\mathcal{P}_k = A \cdot \mathcal{P}_{k-1}$.

Transient Analysis. We now show that $A^k\mathcal{P}_0$ gets exponentially close to the eigenvector $\pi = (\frac{1}{5}, \frac{3}{10}, \frac{1}{5}, \frac{3}{10})$ of A in a small number of steps independent of D and the value of α.

The eigenvalues of A are $\lambda_1 = 1$ and $\lambda_2 = -1/4$, with the other eigenvalues being zero. We diagonalize the matrix to obtain $\Lambda(1, -\frac{1}{4}, 0, 0) = P^{-1}AP$ where P as usual consists of eigenvectors of A.

Let the eigenvectors corresponding to λ_1 and λ_2 be π and π' respectively. The angle between these two eigenvectors is 78.69 degrees. Let q_1 and q_2 be the projections of $\mathcal{P}_0$ onto the one dimensional spaces spanned by π and π' respectively, and let $q' = \mathcal{P}_0 - q_1 - q_2$. Then $A^k\mathcal{P}_0 = A^k(q_1 + q_2 + q') = q_1 + \lambda_2^k q_2$, since $A_k q' = 0$. Since λ_2 is bounded away from 1, it follows that $||A^k\mathcal{P}_0 - \pi||$ drops exponentially fast in k. Thus our steady state eigenvector π will dominate the behavior of $A^k\mathcal{P}_0$ for even modest values of k.

Corollary 2.7 *On average, for random values of $\alpha \gg |x|$, the* NAF *representation of α has* $2/3$ *of its output digits equal to* 0.

Proof. By Theorem 2.6, out of every ten instances of α_k, we expect two to be $0 \bmod 4$, three to be $1 \bmod 4$, two to be $2 \bmod 4$, and three to be $3 \bmod 4$. Hence we produce on average two values of $g_D(\alpha_k)$ equal to $(0, 0)$, three values of $g_D(\alpha_k)$ equal to $(0, 1)$, two values of $g_D(\alpha_k)$ equal to 0, and three values of $g_D(\alpha_k)$ equal to $(0, x)$. Counting up the digits, we find that on average 12 out of the 18 output digits are equal to 0.

2.2 Generalizations

The techniques described above generalize readily to any digit set $D = \{0, 1\} \cup X$ where X consists of $2^n - 1$ elements belonging to prescribed congruence classes mod 2^n. For example, using $n = 3$ we have been able to construct digit sets with proven 78% asymptotic sparsity (compared with 67% in Corollary 2.7 and 74% in Corollary 2.10). However, as a compromise between readability and generality, and also for space reasons, we limit our analysis here to the case of digit sets having five elements. We consider digit sets of the form $D = \{0, 1, x, y, z\}$ where $x \equiv 3 \pmod 8$, $y \equiv 5 \pmod 8$ and $z \equiv 7 \pmod 8$ are negative. The transition matrix in this case has the same largest and second largest eigenvalues as in the $\{0, 1, x\}$ case, with all other eigenvalues being 0. We emphasize that one has considerable freedom in the design of the transition matrix and that the choices given here merely represent a useful baseline.

Definition 2.8. *For a digit set D of the above form, let $f_D \colon \mathbb{N} \to \mathbb{N}$ and $g_D \colon \mathbb{N} \to D \cup (D \times D)$ be the functions defined by*

$$f_D(n) = \begin{cases} n/8 & n{\equiv}0 \pmod 8 \\ (n-1)/8 & n{\equiv}1 \pmod 8 \\ n/2 & n{\equiv}2 \pmod 8 \\ (n-x)/8 & n{\equiv}3 \pmod 8 \\ n/4 & n{\equiv}4 \pmod 8 \\ (n-y)/8 & n{\equiv}5 \pmod 8 \\ (n-2x)/8 & n{\equiv}6 \pmod 8 \\ (n-z)/8 & n{\equiv}7 \pmod 8 \end{cases}, \qquad g_D(n) = \begin{cases} (0,0,0) & n{\equiv}0 \pmod 8 \\ (0,0,1) & n{\equiv}1 \pmod 8 \\ 0 & n{\equiv}2 \pmod 8 \\ (0,0,x) & n{\equiv}3 \pmod 8 \\ (0,0) & n{\equiv}4 \pmod 8 \\ (0,0,y) & n{\equiv}5 \pmod 8 \\ (0,x,0) & n{\equiv}6 \pmod 8 \\ (0,0,z) & n{\equiv}7 \pmod 8 \end{cases}.$$

Theorem 2.9 *Let $\alpha \in \mathbb{N}$ and $D = \{0, 1, x, y, z\}$ as above. Any* NAF *representation $(b_\ell \cdots b_2 b_1 b_0)_2$ of $f_D(\alpha)$ yields a* NAF *representation $(b_\ell \cdots b_2 b_1 b_0 \mid\mid g_D(\alpha))_2$ of α via concatenation. Furthermore, the the probability distribution of the congruence class of $f_D^k(\alpha) \bmod 8$, for random uniformly selected integers $\alpha \in [0, N]$, where $N \gg |\max(x, y, z)|$, converges to the vector $(\frac{1}{10}, \frac{7}{40}, \frac{1}{10}, \frac{1}{8}, \frac{1}{10}, \frac{7}{40}, \frac{1}{10}, \frac{1}{8})$ as $k \to \infty$, with error bounded in magnitude by an exponential in k.*

Proof. By hypothesis, the initial input α has probability distribution $\mathcal{P}_0 = (\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$ over the congruence classes mod 8. The probability distribution $\mathcal{P}_1$ for $f_D(\alpha)$ is computed as follows:

- Assume that α is uniformly distributed in $[0, N]$.
- If $\alpha \equiv 0 \pmod 8$, then $\alpha_1 = \alpha/8$ is uniformly distributed mod 8.
- If $\alpha \equiv 1 \pmod 8$, then $\alpha_1 = \frac{\alpha-1}{8}$ is uniformly distributed mod 8.
- If $\alpha \equiv 2 \pmod 8$, then $\alpha_1 = \alpha/2$ is uniformly 1 or 5mod 8.
- If $\alpha \equiv 3 \pmod 8$, then $\alpha_1 = \frac{\alpha-x}{8}$ is uniformly distributed mod 8.
- If $\alpha \equiv 4 \pmod 8$, then $\alpha_1 = \alpha/4$ is uniformly 1, 3, 5 or 7 mod 8.
- If $\alpha \equiv 5 \pmod 8$, then $\alpha_1 = \frac{\alpha-y}{8}$ is uniformly distributed mod 8.
- If $\alpha \equiv 6 \pmod 8$, then $\alpha_1 = \frac{\alpha-2x}{8}$ is uniformly distributed mod 8.
- If $\alpha \equiv 7 \pmod 8$, then $\alpha_1 = \frac{\alpha-z}{8}$ is uniformly distributed mod 8.

Denote by B the transition matrix

$$B = \begin{pmatrix} \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{2} & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{pmatrix}.$$

Then $P_1 = B \cdot P_0$, and as in the case of Theorem 2.6, the limit distribution of $\mathcal{P}_k = B^k P_0$ for $f_D^k(\alpha)$ converges exponentially rapidly to the eigenvector $\pi = (\frac{1}{10}, \frac{7}{40}, \frac{1}{10}, \frac{1}{8}, \frac{1}{10}, \frac{7}{40}, \frac{1}{10}, \frac{1}{8})$ of B.

Corollary 2.10 *On average, for random values of $\alpha \gg |x|$, the* NAF *representation of α has $20/27$ of its output digits equal to 0.*

Proof. Counting in the same manner as Corollary 2.7, we find that for every forty instances of α_k, on average 80 out of the 108 output digits are equal to 0.

$\alpha = 111101011000010010110001111001111000011000110001111101010101101000100\backslash$
$00010111101010101110111100011111111010100101111001000000101111001010001111\backslash$
$1101010111011000011100001000111100011101010110011$
$\alpha = z00000y0010010000z00x000x0000z00z00y00000z00y0000100z00000y00x0000x0\backslash$
$0001000100x000x0x000x0z00x00100y0000100y0000y000010000y00y00100x0000x00x\backslash$
$000000100000z00x00y00y00x0001000x0x000x00y0010010111$
for $X = \{0, 1, -709, -947, -913\}$
$\alpha = z00z000100x000000100100000010000y000x000000z0000y001001000x01000x0010\backslash$
$000000100y0000z000x0x00000x0001000y00y00100x00z000x0z00y0000x0000x000010\backslash$
$000x00x000x000000000z0010000100000y0010111010110110011110000001$
for $X = \{0, 1, -152397797, -272310435, -132159113\}$

Fig. 1. Examples of randomized sparse representations of a fixed 192-bit integer α with respect to random digit sets $X = \{0, 1, x, y, z\}$. In each representation, the least significant digits are written on the left.

3 Empirical Results

We begin by describing the standard technique for implementing elliptic curve scalar multiplication using non-adjacent form representations. Let α be an integer having a NAF representation $\alpha = (a_k \cdots a_1 a_0)$ with respect to some digit set $D = \{0, 1, x, y, z, \ldots\}$. Compute the point xQ (and also yQ, zQ etc. if needed). The computation of xQ is very fast if $|x|$ is small, and even for large values of $|x|$ there are some protocols (such as ElGamal encryption) for which the point Q is fixed, in which case xQ may be precomputed and stored. One can then compute $\alpha Q = \sum 2^k (a_k Q)$ using the usual double and add formula except with xQ (resp. yQ, zQ) in place of Q whenever the representation of d contains an x (resp. y, z) term as opposed to a 1 term. The efficiency of this calculation depends in large part on the proportion of terms in the representation which are nonzero, since these are the terms that trigger addition operations in the standard double and add formula.

In order to make this algorithm practical for random digit sets, we need to allow the use of integer representations which lack the non-adjacency property, since not every integer has a NAF representation with respect to every digit set. Without this allowance, the algorithm $\mathcal{A}$ would enter into an infinite loop when presented with input values α that lack NAF representations. Our approach is to revert to standard binary representation whenever the algorithm $\mathcal{A}$ encounters an input value of size less than that of one of the digits in the digit set. In this case, the maximum possible length of the ensuing purely binary portion is $\ell := \log(\max\{|x|, |y|, |z|, \ldots\})$. Hence, for $\alpha \gg \max\{|x|, |y|, |z|, \ldots\}$, the statistical analysis of the previous section remains valid for the $1 - \frac{\ell}{\log|\alpha|}$ fraction of the digit string which comprises the vast majority of the representation of α.

Figure 1 contains examples of a 192-bit integer represented in random sparse format with respect to various digit sets $\{0, 1, x, y, z\}$. Figure 2 compares the measured performance of the randomized exponentiation algorithm versus signed

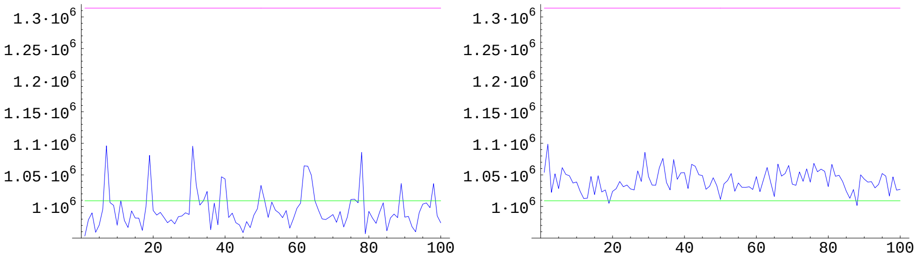

Fig. 2. Empirical timings for 192-bit EC exponentiation implemented on a 3.0 GHz Pentium 4 processor using the Microsoft bignum library. The vertical axis represents clock cycles and the horizontal axis depicts the results of 100 trials. In the left graph, each trial took place using a randomly selected digit set $\{0, 1, x, y, z\}$ with 10-bit values for x, y, z; the right graph uses 32-bit values. In each graph the two horizontal lines represent the cycle count for standard and signed binary representation, respectively.

binary exponentiation as well as standard double-and-add exponentiation. On average, the randomized algorithm outperforms signed binary multiplication for values of x, y, z as large as 10 bits, and remains competitive at 32-bit values. The timings do not include the cost of computing the individual multiples xQ, yQ, zQ, but in performance contexts this cost can be minimized by selecting small values for x, y, z. In the next section, however, we consider digit set randomization in the setting of side channel attacks, and in this setting we do need to use large values of x, y, z and account for the ensuing computational cost.

4 Digit Set Randomization as a Side Channel Attack Countermeasure

Side channel attacks [11] remain one of the most critical points of vulnerability for elliptic curve cryptosystem implementations as they exist today. These attacks make use of power consumption, cache hit rate, timing, or other differences between EC add and EC double operations to determine the binary representation of a scalar multiplier in an EC exponentiation operation [9]. While a number of protective countermeasures against side channel attacks have been proposed ([4, 6, 8, 12, 13, 20, 26, 28]; see [3] for overview), many of the schemes have been broken [5, 16–18, 21, 24, 25, 27] owing to their ad-hoc nature, and all of the existing proposals involve significant performance penalties.

We make a distinction between two classes of side channel attacks known as *simple* and *differential* attacks. In simple side channel attacks, a single execution instance is analyzed and the secret key is deduced using side channel information from that instance alone. In differential attacks, side channel information from multiple execution instances are compared and processed to deduce the secret key. For obvious reasons, it is generally considered more difficult to protect against differential attacks than against simple attacks. In this section we explain how randomized digit sets can be used to leverage simple side channel resilience into differential side channel resilience and present a formal analysis of security under a simplified attack model. Our goal here is not to provide a comprehensive proof of security, but rather just to suggest a new and promising type of approach which has never been considered before, and propose a preliminary naive security analysis as motivation.

A typical side channel attack operates by using side channel information to deduce the internal representation of a secret multiplier, for example by exploiting differences in power consumption between the main branches of a multiplication algorithm. In most cases, knowing the internal representation of an integer is enough to deduce the value of the multiplier. However, when randomized digit sets are used, a given internal (symbolic digit) representation can correspond to a multitude of different integer values, depending on which digit set is used. Hence, even if an attacker possesses full knowledge of the symbolic representation of an integer, we can still quantify to what extent does the value of the integer remain uncertain. Formally, we define the *open representation model* to denote the attack model in which the attacker possesses no information other than the symbolic digits corresponding to the secret multiplier α, and ask how many bits of information entropy remain in the value of α. In the next section we analyze this question and show that the number of bits is equal to the entropy of the digit set, assuming that this entropy is itself less than the entropy of α.

The computation of the individual multiples xQ, yQ, zQ in the RSF algorithm must be done in a side channel resistant manner in order to prevent the attacker from determining the values of x, y, z via side channel analysis. However, since x, y, z are randomly selected at runtime, the computation of xQ will only be performed once for any given value of x, and thus this computation only needs to resist simple side channel attacks.

Although some aspects of the open representation model lack realism—for example, a real attacker would likely know the value of αQ—we believe that the model is useful because it isolates the effects of side channel leakage in a well defined way. Our introduction of this attack model is novel since other side channel countermeasures do not make the crucial distinction between symbolic representations and integer values which is necessary in order for the model to be non-vacuous. Indeed, most side channel countermeasures in the literature rely on manipulating either the representation itself or the sequence of field operations used, and do not provide any security under the open representation model.

5 Entropy Bounds on Randomized Representations of Integers

In order to evaluate the security of digit set randomization in the open representation model, we now determine for a given digit string how many random digit sets D will produce a fixed number α under that digit string. In order to avoid the awkward issue of how to select random digit sets out of an infinite collection, we assume that the elements of D are bounded in absolute value by some fixed bound (such as 2^{32}) which is very small relative to α. We also assume for simplicity that the elements of D are all negative except for 0 and 1. However, we emphasize that this analysis does not depend on the NAF property or indeed any other property of the digit string in question. Our analysis uses the Gaussian Heuristic [23] for lattices which states in any well behaved subset of $\mathbb{R}^n$ the number of lattice points inside is approximated by the ratio of the volume of the body to the lattice determinant.

5.1 One Random Term in D

The simplest case of randomized digit sets is sets of the form $D = \{0, 1\} \cup X$ where $X = \{x\}$, $x < 0$ is randomly selected (possibly under some mild constraints, such as $x \equiv 3 \pmod 4$, whose effect will be explained below). In this case, given a digit string $(a_k \cdots a_2 a_1 a_0)_2$, the corresponding value of α is

$$\alpha = \sum_{i=0}^{k} a_i 2^i = A_0 + A_1 x, \qquad \text{where } A_0 = \sum_{a_i=1} 2^i, \quad A_1 = \sum_{a_i=x} 2^i.$$

For any given value of $\alpha \gg |x|$, there is only one value of x that will satisfy the equation $A_0 + A_1 x = \alpha$. Thus the information entropy of α is exactly equal to the entropy of X.

The condition $x \equiv 3 \pmod 4$ means that an attacker who obtains the complete representation of α can obtain the two least significant bits of α using the formula $\alpha = A_0 + A_1 x$. This phenomenon can also be seen in Figure 1 where the last two digits (or three digits, in the case of digit sets defined $\bmod 8$) of the representation are independent of the digit set. However, this level of information leakage must be put into perspective: without digit set randomization, the entire integer α would already be known, as opposed to two or three bits.

5.2 Two Random Terms in D

If we consider digit sets $D = \{0, 1\} \cup X$ where $X = \{x, y\}$, then we have

$$\alpha = \sum_{i=0}^{k} a_i 2^i = A_0 + A_1 x + A_2 y, \text{ where } A_0 = \sum_{a_i=1} 2^i, A_1 = \sum_{a_i=x} 2^i, A_2 = \sum_{a_i=y} 2^i.$$

For fixed $\alpha, A_0, A_1, A_2 > 0$, the number of negative integer solutions (x, y) to $\alpha = A_0 + A_1 x + A_2 y$ (or, equivalently, the number of positive integer solutions (x, y) to $A_0 - \alpha = A_1 x + A_2 y$) is bounded above by

$$\frac{(A_0 - \alpha)}{A_1 A_2} \cdot \gcd(A_1, A_2).$$

This bound is obtained using standard linear Diophantine analysis. For convenience, we sketch the argument here. Let $Ax + By = C$ be a linear Diophantine equation in two variables, with $A, B, C > 0$. Divide out by $\gcd(A, B)$ to obtain a new equation $ax + by = c$ with $\gcd(a, b) = 1$. If (x_0, y_0) is one solution to the equation, then all solutions to the equation must have the form $(x, y) = (x_0 + bt, y_0 - at)$, where t is an integer parameter. If we require x and y to be positive, then that imposes the bounds $0 < x < c/a$ on x, and the number of integers of the form $x = x_0 + bt$ that satisfy $0 < x < c/a$ is upper bounded by

$$\frac{c/a}{b} = \frac{c}{ab} = \frac{C}{AB} \cdot \gcd(A, B),$$

as desired. In particular, in expectation one would get $(A_0 - \alpha) = \Theta(\alpha) = \Theta(A_1) = \Theta(A_2)$ and $\gcd(A_1, A_2) = O(1)$, so $(A_0 - \alpha)\gcd(A_1, A_2) = \Theta(\alpha)$ is overwhelmingly likely to be less than $A_1 A_2 = \Theta(\alpha^2)$. Therefore, on average, we expect at most one negative integer solution (x, y) to the equation $\alpha = A_0 + A_1 x + A_2 y$, and thus the information entropy in computing α for the attacker is equal to the entropy in computing x and y.

5.3 General Case

In general, with $D = \{0, 1\} \cup X$, where $X = \{x_1, x_2, \ldots, x_c\}$, we find that the corresponding Diophantine equation $\alpha = A_0 + \sum_{i=1}^{c} A_i y_i$ has

$$\frac{1}{(c-1)!} \cdot \frac{(A_0 - \alpha)^{c-1}}{\prod_{i=1}^{c} A_i} \cdot \gcd(A_1, \ldots, A_c)$$

negative integer solutions. Here the numerator has approximate size $O(\alpha^{c-1})$ and the denominator $O(\alpha^c)$, so on average each α will have at most one negative integer solution.

6 Conclusions and Further Work

We present a method for using randomized digit sets in integer representations and give empirical results showing that elliptic curve point multiplication algorithms based on large randomized digit sets outperform both standard and signed binary representations. Our theoretical analysis of the sparsity of randomized digit set representations simplifies and generalizes the existing analyses available in the literature. We also propose digit set randomization as a side channel attack countermeasure, and provide a preliminary analysis of the security of random digit sets under a new attack model called the open representation model which is designed to isolate the impact of side channel information leakage. Our randomized algorithm is one of the only side channel countermeasures available that achieves even some level of security under this attack model.

In this paper we have not yet made any attempt to find parameters for digit set randomization which both simultaneously achieve good performance and good side channel resilience. In the future, we hope to perform empirical trials comparing the performance of random digit sets with various parameters against other existing side channel countermeasures; this task is greatly complicated by the large number and variety of side channel attack countermeasures which have been proposed. However, based on the fact that performance-oriented choices of digit set parameters lead to record or near record levels of performance, we are optimistic that digit set randomization provides a good foundation for future work towards high performing side channel attack resistant algorithms.

Acknowledgments. We are grateful to James Muir for his helpful and detailed suggestions to us during the preparation of this manuscript.

References

1. G. Avione, J. Monnerat, and T. Peyrin, *Advances in Alternative Non-adjacent Form Representations*, Advances in cryptology—Indocrypt 2004, Lecture Notes in Computer Science, vol. 3348, Springer, Berlin, 2004, pp. 260–274.
2. W. Bosma, *Signed bits and fast exponentiation*, J. Théor. Nombres Bordeaux **13**, no. 1, 27–41 (English, with English and French summaries). 21st Journées Arithmétiques (Rome, 2001).
3. H. Cohen and G. Frey, *Handbook of elliptic and hyperelliptic curve cryptography*, Discrete Mathematics and its Applications (Boca Raton), Chapman & Hall/CRC, Boca Raton, FL, 2006.
4. J.S. Coron, *Resistance against differential power analysis for elliptic curve cryptosystems*, Cryptographic hardware and embedded systems—CHES 1999, Lecture Notes in Computer Science, vol. 1717, Springer, Berlin, 1999, pp. 231–237.
5. P.A. Fouque, F. Muller, G. Poupard, and F. Valette, *Defeating countermeasures based on randomized BSD representation*, Cryptographic Hardware and Embedded Systems—CHES 2004, Lecture Notes in Computer Science, vol. 3156, Springer, Berlin, 2004, pp. 312–327.
6. J.C. Ha and S.J. Moon, *Randomized Signed-Scalar Multiplication of ECC to Resist Power Attacks*, Cryptographic Hardware and Embedded Systems—CHES 2002, Lecture Notes in Computer Science, vol. 2523, Springer, Berlin, 2002, pp. 551–563.
7. C. Heuberger and H. Prodinger, *Analysis of alternative digit sets for nonadjacent representations*, Monatsh. Math. **147** (2006), no. 3, 219–248.
8. K. Itoh, J. Yajima, M. Takaneka, and N. Torii, *DPA countermeasures by improving the window method*, Cryptographic hardware and embedded systems—CHES 2002, Lecture Notes in Comput. Sci., vol. 2523, Springer, Berlin, 2002, pp. 303–317.
9. J. Jaffe, B. Jun, and P. Kocher, *Differential Power Analysis*, Advances in cryptology—CRYPTO '99, Lecture Notes in Computer Science, vol. 1666, Springer, Berlin, 1999, pp. 388–397.
10. M. Joye and S.-M. Yen, *Optimal left-to-right binary signed-digit recoding*, IEEE Trans. on Computers **49** (2000), no. 7, 740–748.
11. P. Kocher, *Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems.*, Advances in cryptology—CRYPTO '96, Lecture Notes in Computer Science, vol. 1109, Springer, Berlin, 1996, pp. 104–113.
12. P.-Y. Liardet and N.P. Smart, *Preventing SPA/DPA in ECC systems using the Jacobi form*, Cryptographic hardware and embedded systems—CHES 2001, Lecture Notes in Comput. Sci., vol. 2162, Springer, Berlin, 2001, pp. 391–401.

13. B. Möller, *Securing Elliptic Curve Point Multiplication against Side-Channel Attacks*, Information Security Conference (ISC 2001), Lecture Notes in Comput. Sci., vol. 2200, Springer, Berlin, 2001, pp. 324–334.
14. F. Morain and J. Olivos, *Speeding up the computations on an elliptic curve using addition-subtraction chains*, RAIRO Inform. Théor. Appl. **24** (1990), no. 6, 531–543 (English, with French summary).
15. J.A. Muir and D.R. Stinson, *Alternative digit sets for nonadjacent representations*, SIAM J. Discrete Math. **19** (2005), no. 1, 165–191.
16. K. Okeya and D.-G. Han, *Side Channel Attack on Ha-Moon's Countermeasure of Randomized Signed Scalar Multiplication*, Advances in cryptology—Indocrypt 2003, Lecture Notes in Comput. Sci., vol. 2904, Springer, Berlin, 2003, pp. 334–348.
17. K. Okeya and K. Sakurai, *A Second-Order DPA Attack Breaks a Window-method based Countermeasure against Side Channel Attacks*, Information Security Conference (ISC 2002), pp. 389–401.
18. ———, *On Insecurity of the Side Channel Attack Countermeasure using Addition-Subtraction Chains under Distinguishability between Addition and doubling*, 7th Australasian Conference on Information Security and Privacy (ACISP 2002), Lecture Notes in Comput. Sci., vol. 2384, Springer, Berlin, 2002, pp. 420–435.
19. D.A. Osvik, A. Shamir, and E. Tromer, *Cache Attacks and Countermeasures: The Case of AES (Extended Version)*, Proceedings of ICISC 2004, Lecture Notes in Computer Science, vol. 3506, Springer, Berlin, 2004, pp. 154–167.
20. E. Oswald and M. Aigner, *Randomized addition-subtraction chains as a countermeasure against power attacks*, Cryptographic hardware and embedded systems—CHES 2001 (Paris), Lecture Notes in Comput. Sci., vol. 2162, Springer, Berlin, 2001, pp. 39–50.
21. D.J. Park and P.J. Lee, *A DPA Attack on the Improved Ha-Moon Algorithm*, Workshop on Information Security Applications (WISA 2005), Lecture Notes in Computer Science, vol. 3786, Springer, Berlin, 2006, pp. 283–291.
22. G.W. Reitwiesner, *Binary arithmetic*, Advances in computers, Vol. 1, 1960, pp. 231–308.
23. C. P. Schnorr and H. H. Hörner, *Attacking the Chor-Rivest cryptosystem by improved lattice reduction*, Advances in cryptology—EUROCRYPT '95, Lecture Notes in Comput. Sci., vol. 921, Springer, Berlin, pp. 1–12.
24. S.G. Sim, D.J. Park, and P.J. Lee, *New power analyses on the Ha-Moon algorithm and the MIST algorithm*, ICICS 2004, Lecture Notes in Comput. Sci., vol. 3269, Springer, Berlin, 2004, pp. 291–304.
25. C.D. Walter, *Breaking the Liardet-Smart randomized exponentiation algorithm*, Smart Card Research and Advanced Applications—CARDIS 2002, Usenix Association.
26. ———, *Some security aspects of the MIST randomzied exponentiation algorithm*, Cryptographic Hardware and Embedded Systems—CHES 2002, Lecture Notes in Computer Science, vol. 2523, Springer, Berlin, 2002, pp. 276–290.
27. ———, *Issues of Security with the Oswald-Aigner Exponentiation Algorithm*, Topics in cryptology—CT-RSA 2004, Lecture Notes in Comput. Sci., vol. 2964, Springer, Berlin, 2004, pp. 208–221.
28. S.M. Yen, C.N. Chen, S. Moon, and J. Ha, *Improvement on Ha-Moon randomized exponentiation algorithm*, Proceedings of ICISC 2004, Lecture Notes in Computer Science, vol. 3506, Springer, Berlin, 2004, pp. 154–167.