Random Digit Representation of Integers

Nicolas Méloni* and M. Anwar Hasan†

June 23, 2015

Abstract

Modular exponentiation is core to today's main stream public key cryptographic systems. In this article, we generalize the classical fractional wNAF method for modular exponentiation – the classical method uses a digit set of the form $\{1, 3, \ldots, m\}$ which is extended here to any set of odd integers of the form $\{1, d_2, \ldots, d_n\}$. We give a formula for the average density of non-zero terms in this new representation and discuss its asymptotic behavior when those digits are randomly chosen from a given set. We also propose a specific method for the precomputation phase of the exponentiation algorithm.

1 Introduction

Let $k = (k_{t-1} \ldots k_0)_2$ be an integer and G a group. For $g \in G$, one can always compute g^k with at most $2\log(k) = 2t$ group operations using the classical `square-and-multiply` algorithm (or `double-and-add` in its additive form). There are various methods to speed up the exponentiation process, most of them are based on the initial idea from Brauer [1] that uses the 2^w-ary representation of k and performs the exponentiation accordingly. Generally speaking, those methods consider a recoding of k of the form $\sum_{i=0}^{t-1} k_i 2^i$ with k_i in some digit set D and performs the exponentiation with an adapted version of the `square-and-multiply` algorithm. Many improvements have been proposed over the years, from signed digits to sliding and fractional windows [17, 14]; see [2, 7] for a general overview. Common to those improvements is the use of digit sets containing odd integers lower than some fixed bound. In the present work we propose to generalize those approaches to any set of digits. We propose a general recoding algorithm using any digit set containing 1 and give a general formulae to compute the average density of non-zero terms of the recoding. All these lead to a new randomized exponentiation scheme that can be used as a countermeasure to power analysis. It has the advantages of providing more randomness than classical randomization techniques and being at the same time almost as asymptotically efficient as standard exponentiation methods.

The rest of this paper is organized as follows: Section 2 is a brief review of the standard fractional window exponentiation method, in Section 3 we describe our new recoding algorithm and give a formulae to compute its average density of non-zero terms for any set of digits, in Section 4 we describe our new randomized exponentiation scheme, study its average density and propose a specific method for the precomputation phase of the exponentiation and in Section 5 we discuss the security provided by that new scheme and propose some comparisons with previous related works.

2 Preliminaries

2.1 Fast exponentiation

Let k, g and G be as defined above. Most standard fast exponentiation schemes fall into a general framework. First find a recoding of $k = (k_{l-1} \ldots k_0)_2$ with $k_i \in \mathcal{D} \cup \{0\}$, for some set $\mathcal{D}$. Then compute g^k using Algorithm

*N. Méloni is with the Université de Toulon. nicolas.meloni@univ-tln.fr

†M. A. Hasan is with the Department of Electrical and Computer Engineering of the University of Waterloo.

1. From this perspective, successive improvements of the `square-and-multiply` algorithm can be viewed as a new recoding scheme using larger and larger sets of digits and providing sparser and sparser recoding, that is with as few non-zero terms as possible. All put together, the various improvements give the *fractional windows* or *frac w-NAF* proposed by Möller [14]. In that case, $\mathcal{D} = B_m = \{\pm 1, \pm 3, \ldots, \pm m\}$. Möller proved that the average density of non-zero terms of this representation is $\frac{1}{a+1}$, where $a = W_n + \frac{1+m}{2^{W_n-2}} - 2$ and $W_n = \lfloor \log_2 k \rfloor + 2$. In this work we generalize this approach to any set $\mathcal{D}$ containing 1.

Algorithm 1 Computation of g^k

Require: An integer $k = (k_{l-1} \ldots k_0)_2$, an element g and a set of integers $\mathcal{D}$
Ensure: g^k
1: $h \leftarrow 1$
2: **for** $d \in \mathcal{D}$ **do**
3: $\quad G_d \leftarrow g^d$
4: **end for**
5: **for** $i = l-1 \ldots 0$ **do**
6: $\quad h \leftarrow h^2$
7: $\quad$ **if** $k_i \neq 0$ **then**
8: $\quad\quad h \leftarrow h \times G_{k_i}$
9: $\quad$ **end if**
10: **end for**
11: **return** h

Remark 1. *In certain contexts, it has been proven to be more efficient to consider recoding using a different base than 2. For instance, fast cubing can lead to consider ternary or hybrid binary/ternary representations [3] and fast group endomorphism have been used in producing complex representations such as the τNAF on Koblitz curves [12].*

3 Random digit representation

Let $\mathcal{D} = \{\pm d_1, \ldots, \pm d_l\}$ be a set of odd integers. We call $\mathcal{D}$-representation of k any recoding of the form $k = \sum k_i 2^i$ with $k_i \in \overline{\mathcal{D}} \cup \{0\}$. We define $N(\mathcal{D})$ as the set of all integers for which there exists a $\mathcal{D}$-representation. It is clear that any integer in $N(\mathcal{D})$ is a multiple of the gcd of $\mathcal{D}$. Thus, in order to have $N(\mathcal{D}) = \mathbb{Z}$ we must have $\gcd(\mathcal{D}) = 1$. However the reverse does not hold. Indeed, with $\mathcal{D} = \{\pm 5, \pm 13\}$, 1 does not have a $\mathcal{D}$-representation. On the other hand, as long as 1 belongs to $\mathcal{D}$ we are guaranteed that $N(\mathcal{D}) = \mathbb{Z}$ since its binary representation is a $\mathcal{D}$-representation for any k. In the rest of the paper, we will only consider sets on the form $\{\pm 1, \pm d_2, \ldots, \pm d_n\}$.

3.1 The recoding algorithm

Let us start with a few notations. Let $w > 0$ be an integer. For any integer x we define $p_w(x) := x \mod 2^w$. We then set $\mathcal{D}_w = p_w(\mathcal{D})$ and $\overline{\mathcal{D}}_w = \mathcal{D}_w \cup \{2^w - d : d \in \mathcal{D}_w\}$. Finally, we define $w_n = \lfloor \log_2(\max_i(d_i)) \rfloor + 1$ and $W_n = w_n + 1$.

In order to define the recoding map, we first need to define, for all integer k, $w_{max}(k)$ as the largest integer $w \leq W_n$ such that there exists a digit $d_i \in \mathcal{D}$ satisfying the following two conditions:

1. $d_i < k$,
2. $p_w(k) = p_w(d_i)$ or $2^w - p_w(k) = p_w(d_i)$.

Finally, let the mapping $digit_{\mathcal{D}} : \mathbb{N} \to \overline{\mathcal{D}} \cup \{0\}$ be defined as follows:

- set $W_{max} = w_{max}(k)$

- if k is even, $digit_{\mathcal{D}}(k) = 0$,
- if $p_{W_{max}}(k) \in \mathcal{D}_{W_{max}}$, $digit_{\mathcal{D}}(k) = d$ with any integer $d < k$ such that $p_w(k) = p_w(d)$
- if $2^{W_{max}} - (k \mod 2^{W_{max}}) \in \mathcal{D}_{W_{max}}$, $digit_{\mathcal{D}}(k) = -d$ with any integer $d < k$ such that $2^w - p_w(k) = p_w(d_i)$.

It is important to remark that the map is well defined, that is to say that $digit_{\mathcal{D}}(k)$ exists for all k. Indeed $1 \in \mathcal{D}_w$ for any w. The following algorithm uses the $digit_{\mathcal{D}}$ map to compute the $\mathcal{D}$-representation of any given k.

Algorithm 2 Random Digit Representation of integer k

Require: An integer k and a set $\mathcal{D} = \{\pm 1, \pm d_2, \ldots, \pm d_n\}$
Ensure: $k = (k_t k_{t-1} \ldots k_1 k_0)_2 \in N(\mathcal{D})$
1: $i = 0$
2: **while** $k! = 0$ **do**
3: $\quad k_i = digit_{\mathcal{D}}(k)$
4: $\quad k = \frac{k - k_i}{2}$
5: $\quad i = i + 1$
6: **end while**
7: **return** $(k_{i-1} \ldots k_0)$

Remark 2. *If $\mathcal{D} = \{\pm 1, \ldots, \pm m\}$ we obtain exactly the* fractional windows *recoding.*

Example 1. *Let $k = 31415$ and $\mathcal{D} = \{1, 3, 23, 27\}$. We have $\mathcal{D}_2 = \{1, 3\}$, $\mathcal{D}_3 = \{1, 3, 7\}$, $\mathcal{D}_4 = \{1, 3, 7, 11\}$ and $\mathcal{D}_5 = \mathcal{D}_6 = \mathcal{D}$. Algorithm 2 applied to k gives:*

1. *k is odd, $k \mod 2^6 \equiv 55 \equiv -19$ and $k \mod 2^5 \equiv 23$, so $w_{\max} = 5$ and $digit_{\mathcal{D}}(k) = 23$,*
2. *$k_0 = 23$ and $k = \frac{k-23}{2} = 15696$.*
3. *$k_1 = k_2 = k_3 = k_4 = 0$ and $k = \frac{k}{2^4} = 981$.*
4. *k is odd, $k \mod 2^6 \equiv 21 \equiv -43$, $k \mod 2^5 \equiv 21 \equiv -11$ and $k \mod 2^4 \equiv 5 \equiv -11$ thus $w_{\max} \equiv 4$ and $digit_{\mathcal{D}}(k) = -p_5^{-1}(11) = -27$*
5. *$k_5 = -27$ and $k = \frac{k+27}{2} = 504$,*
6. *$k_6 = k_7 = k_8 = 0$ and $k = k/8 = 63$,*
7. *$k \mod 2^6 \equiv -1$ so $w_{max} = 6$ and $k_9 = -1$*
8. *$k_{10} = k_{11} = k_{12} = k_{13} = k_{14} = 0$, $k = \frac{k+1}{2^6}$*
9. *$k = k_{15} = 1$.*

Finally we obtain $k = (1, 0, 0, 0, 0, 0, -1, 0, 0, 0, -27, 0, 0, 0, 0, 23)_2$.

3.2 Average density

Theorem 1. *Let k be an integer and $\mathcal{D} = \{1, d_2, \ldots, d_n\}$ a set of digits. Then the asymptotic average density of non-zero terms achieved by the random digit is $\frac{1}{a+1}$, where*

$$a = 2D(W_n) + D(W_n - 1) + D(W_n - 2) + \cdots + D(2)$$

and $D(w) = \frac{\#\overline{\mathcal{D}}_w}{2^{w-1}}$.

Proof. Let k be an odd integer greater than d_n and $d = digit_{\mathcal{D}}(k)$. We first want to evaluate the probability $P(w)$ that $w_{max}(k) = w$ for every $w \leq W_n$. Let $D(w)$ be the probability that the residue of a given odd integer modulo 2^w lies in $\overline{\mathcal{D}}_w$. By construction we have that $P(W_n) = D(W_n)$. Moreover, it is clear that $d \in \overline{\mathcal{D}}_w \Rightarrow \overline{\mathcal{D}}_{w-1}$ which implies that, for $w < W_n$, $P(w) = D(w) - D(w+1)$.

Now, from the definition of $w_{max}(k)$, we have $(k - d) \equiv 0 \mod 2^{w_{max}}$. In other words, k can be written in the form

$$k = (k'_{t'} \dots k'_{w_{max}} 0 \dots 0 d)_2, k_i \in 0, 1.$$

If $w_{max} < W_n$, still by definition, $k'_{w_{max}} \neq 0$. However, if $w_{max} = W_n$ it is necessary to estimate the average number of consecutive zeros starting from $k'_{w_{max}}$. Classically, for an arbitrary long sequence of random bits, this number is 1. As a consequence, we can finally write the value of a, the average number of zeros following a non-zero digit in the RDR of an integer in terms of $D(w)$:

$$\begin{aligned} a &= (W_n)P(W_n) + (W_n - 2)P(W_n - 1) + \dots + 2P(3) + P(2) \\ &= (W_n)D(W_n) + (W_n - 2)(D(W_n - 1) - D(W_n)) + \dots + D(2) - D(3) \\ &= 2D(W_n) + D(W_n - 1) + \dots + D(3) + D(2) \end{aligned}$$

□

Example 2. *For set $B_m = \{1, 3, \dots, m\}$ we have $w_n = \lfloor \log_2(m) \rfloor$ and $W_n = w_n + 1$. It is easy to see that, on one hand $D(w_n) = D(w_n - 1) = \dots = D(2) = 1$ and on the other hand $D(W_n) = (m+1)/2^{w_n}$, so that*

$$a = W_n - 2 + \frac{m+1}{2^{w_n - 1}},$$

which corresponds to the standard result on the average density of the frac-wNAF representation.

Example 3. *With $\mathcal{D} = \{1, 3, 23, 27\}$, we have $\overline{\mathcal{D}}_2 = \{1, 3\}$, $\overline{\mathcal{D}}_3 = \{1, 3, 5, 7\}$, $\overline{\mathcal{D}}_4 = \{1, 3, 5, 7, 9, 11, 13, 15\}$, $\overline{\mathcal{D}}_5 = \{1, 3, 5, 9, 23, 27, 29, 31\}$. From Theorem 1 we obtain that*

$$a = 2 \times \frac{1}{4} + \frac{1}{2} + 1 + 1 + 1 = 4.$$

In this case the RDR has a density of $\frac{1}{5}$, the same as the 4-NAF representation, where $\mathcal{D} = \{1, 3, 5, 7\}$. In other words, we see that we can achieve the same density with different sets of digits of the same cardinal.

4 Randomized exponentiation scheme

Algorithm 2 allows us to compute the RDR of an integer k using any set of digits $\mathcal{D}$ as long as 1 belongs to it. We can now integrate this algorithm into a general randomized exponentiation scheme. Let g be a group element, k an exponent and m and t two integers satisfying $t \leq m$. One can compute g^k using the following scheme:

1. randomly choose $t - 1$ odd integers $\{d_2, \dots, d_t\}$ among $\{3, \dots, m\}$,
2. compute RDR of k using Algorithm 2 and set $\mathcal{D} = \{1, d_2, \dots, d_t\}$,
3. compute g^k using Algorithm 1.

One obvious advantage of such a scheme is that it is naturally resistant to differential power analysis. Indeed, from one exponentiation to the other, it ensures that the operation flow will be completely different.

4.1 Average case and the urn problem

From Theorem 1 we can compute the asymptotic density of the RDR for a given set of digits $\mathcal{D}$. One natural question is what is the average behavior of this density when the digits are chosen randomly. Let m be a parameter and let us consider $B_m = \{1, 3, \ldots, m\}$. We want to evaluate the average density of the RDR when we randomly choose l integers from B_m. To apply our theorem we need to compute the value of $D(w)$ for all needed w and thus the cardinal of $\overline{\mathcal{D}_w}$.

First we note that by definition all $d \in \mathcal{D}$ are smaller than 2^{W_n-1}, which implies that $(2^{W_n} - d) \neq d$ and thus $\overline{\mathcal{D}_{W_n}} = 2l$. Evaluating $D(w)$ for smaller value of w becomes a harder problem. If an integer d is in $\mathcal{D}$, then all integer of the form $d + i2^w$ and $i2^w - (d \mod 2^w)$ do not add up anything to the cardinal of $\overline{\mathcal{D}_w}$. In short, we need to evaluate the number of equivalence classes of the set $\mathcal{D}$ with respect to the relation $\mathcal{R}_w$ define for all $w \leq W_n$ by:

$$x \mathcal{R}_w y \Leftrightarrow x \equiv y \mod 2^w, \text{ or } x \equiv -y \mod 2^w.$$

An easy way to consider this problem is to see it as an urn problem. Let us consider $N = \frac{m+1}{2}$ balls corresponding to our initial integer set B_m. For a given w, define $C_w = 2^{w-2}$ as the number of equivalence classes with respect to $\mathcal{R}_w$. For instance, with $w = W_n - 1$, there exist 2^{W_n-2} possible classes (that is odd integer lower than 2^{W_n-1}), each pair $(i \mod 2^{W_n}, 2^{W_n} - (i \mod 2^{W_n}))$ for i in $\{1, 3, \ldots, 2^{W_n-2}\}$ being one of them. Finally, define $E_w^i =$ as the number of elements of class i. Our problem consists in drawing l balls (without replacement) in an urn containing N balls of C_w different colors and evaluate the average number of different colors obtained. Let $M(l, c, N)$ be the number of different drawings, without replacement, of l balls among N having exactly c different colors. Then the probability that we obtain exactly c colors is

$$P[X = c] = \frac{M(l, c, N)}{\binom{N}{l}},$$

where X is a random variable corresponding to the number of drawn colors. A theorem from Walton [20] shows that $M(l, c, N)$ can be computed by developing the polynomial

$$F_w(X, Y) = \prod_{c=1}^{C_w} \left(Y\{(1 + X)^{E_w^i} - 1\} + 1 \right).$$

Indeed, he proves that

$$F_w(X, Y) = \sum_c \sum_l M(l, c, N) X^l Y^c.$$

It is then possible to compute $D(w)$ for practical values of w and finally obtain the density of the RDR representation for a given number of drawings. In this work we have computed it for $w \leq W \leq 10$. Results are summarized in Table 4.2. In order to maximize the number of possible digit sets, we fix $t = \left\lfloor \frac{1+m}{4} \right\rfloor$ as the number of drawn balls. The wNAF column corresponds to the (optimal) density of the wNAF representation using as many digits as the RDR. We observe that the difference between the two methods is relatively small. The general loss in terms of density is less than half a bit.

4.2 Precomputation scheme

The first step of our exponentiation scheme consists of computing $g^{\pm d_i}$ for $d_i \in \mathcal{D}$. Finding an efficient way to perform this computation is somehow equivalent to finding a short addition chain computing the set $\mathcal{D}$. The problem is trivial when $\mathcal{D} = B_m$ as the chain $(1, 2, 3, 5, 7, 9, \ldots, m)$ is the shortest possible. However when the d_i's are randomly chosen it is harder to find an optimal chain. The naive approach consists of using the previous chain and only keep the needed elements. It requires the computation of $m/2$ integers when only $m/4$ could be needed in the best case. Here we propose a method to find shorter addition chains than the naive approach, inspired by Pippinger algorithm [16]. It is a very general algorithm that allows

m	RDR	wNAF
7	3.833	4
15	4.771	5
31	5.728	6
63	6.706	7
127	7.695	8
255	8.689	9
511	9.686	10
1023	10.69	11

Figure 1: Inverses of the density of the RDR and wNAF using $\lfloor \frac{m+1}{4} \rfloor$ digits

the computation of multiple powers of a group element. Our case however does not require such a general method. In particular, we know that

- we need to compute the g^{d_i}'s for small values of d_i,
- all d_i's are odd,
- the cardinal of $\mathcal{D}$ is fixed to $\lfloor \frac{m+1}{4} \rfloor$.

Thus we can use a more simple method described next. Let $0 < b < W$ be a parameter and define $q = \lfloor \frac{m}{2^b} \rfloor$:

1. compute $X = \{g, g^2, g^3, g^5, g^7, \ldots, g^{2^b-1}\}$,
2. compute $Y = \{g^{2^b}, g^{2 \cdot 2^b}, , g^{3 \cdot 2^b}, \ldots, g^{q \cdot 2^b}\}$,
3. for all $d_i \notin X$, compute $g^{d_i} = xy, (x, y) \in X \times Y$.

The computation cost is 2^{b-1} group operations to compute X, q group operations to compute Y and at most $\#\mathcal{D}$ group operations to obtain the g^{d_i}'s. The total cost is thus bounded by $2^{b-1} + \lfloor \frac{m}{2^b} \rfloor + \#\mathcal{D}$ group operations. In the end, we save many operations in the later stage depending on parameter b. Indeed, the proportion of integer d_i in X is given by $\frac{2^b}{m+1}$. So for instance, with $\frac{m+1}{4}$ randomly chosen numbers, our method saves on average 2^{b-2} group operations.

5 Side-channel Security

The main interest of randomizing the exponentiation process is to provide resistance to side-channel attacks via algorithmic countermeasures. In this section we discuss the security of our method against differential and simple side channel attacks.

5.1 Differential attacks

Differential side-channel attacks aim at finding the secret key by analyzing power traces of several executions of the same computation, depending on that secret. Recent works have proven to be able to defeat various randomization methods such as the Binary Signed Digit randomization [4] or Liardet-Smart randomized algorithm [18] for instance. The main weakness of those methods is little randomness they actually provide despite the apparent variety of recoding they provide. In particular, Fouque et al. stress that such randomization techniques fail because they do not provide a sufficiently large number of possible local internal states and transitions from that states, making them vulnerable to collision attacks. Another important remark is

that those attacks use the facts that the set of digits is known in advance. For instance, the hidden Markov model cryptanalysis used against the Oswald-Aigner randomized exponentiation makes a direct use of the knowledge of the three possible digits $0, 1$ and -1 to produce the probabilistic state machines used in the cryptanalysis [11].

From that perspective, our method is the first to provide two levels of protection against those attacks. First, the fact that the digit set is randomly chosen prevents a traditional attackers to mount any attack previously mentioned as they directly use the fact that the digit set is know in advance. In order to mount an attack, all possible digit sets must be considered and dealt with in parallel. For that matter, the size of the set can be seen as a security parameter. For instance, using an eight digit recoding (seven of them randomly chosen from $\{3, \ldots, 31\}$), we obtain a total of 6435 possible digit sets and more than 3×10^8 for a sixteen digit recoding. On top of that, the recoding algorithm itself provides randomness as when several digits satisfy the appropriate congruence one is chosen at random. It means that for a given digit set, any integer can have many different recodings.

5.2 Simple side-channel attacks

Simple side-channel attacks aim at obtaining information on the secret key using a single trace. Typically, being able to distinguish squarings from multiplications allows an adversary to recover the secret exponent of any exponentiation using the `square-and-multiply` algorithm. From that perspective, a randomization process does not, by itself, provide any protection. To ensure that an algorithm is secure against such attacks, the standard way is to make the computation as regular as possible. It can be done at the algorithm level, using the Montgomery ladder for instance, or at the group algorithmic level, for example by using unified formulae in the context of elliptic curve cryptography or using block atomicity.

Our algorithm clearly will not have a regular behavior, however that does not signifies that it is vulnerable to simple attacks. Indeed, one obviously implements it using one of the previously mentioned arithmetic level countermeasures, but even without them, being able to distinguish between squaring and multiplication does not provide much information on the secret key. Even if the sequence of multiplications and squarings performed by the algorithm is given, one still has to guess which digit has been used at each step. For instance, for a 128-bit security, and therefor a 256-bit exponent, and an eight digit recoding (that is with parameter $m = 31$ in Table 1), there will be on average 44.6 non-zero digits in the recoding corresponding to so many multiplications in the trace. As there are 8 possible choices for each of these multiplications the total number of combinations is roughly $8^{44.6} \sim 2^{133.8}$. This has to be multiplied by the number of possible digits sets (6435). Trying to recover the original key from an exhaustive search would be more difficult than attacking the system itself.

Remark 3. *All previous claims are only based on general considerations. It would be presumptuous to consider our scheme secure and great care must be taken for that matter. Many previous schemes have been claimed to be side-channel resistant and have been broken little time after that. However, because of an extra degree of randomization our approach can potentially provide more protection than those previous methods and can be combined with others in order to, at least, make attackers job harder, for very little computational overhead.*

5.3 Related works

Randomization is a standard way to provide security against differential side-channel attacks. In particular, several randomized exponentiation algorithms have been proposed [6, 15, 10, 19, 8] but the security offered by those methods remains in general uncertain. For example, randomized recodings proposed by Ha and Moon [6] or Oswald and Aigner [15] have been defeated due to little local variation of the data. It was exploited through collision detection [4] or more generally using the hidden Markov model cryptanalysis [11]. In a similar way, some randomization techniques focus on the management of the window in sliding window algorithm [9, 13] and successful attacks have been mounted against some of them [18]. More generally, the

hidden Markov model attack seems to be a threat to all of them. Finally, very recently Guérini, Imbert and Winterhalter proposed a new recoding method based on exact covering systems of congruences [5]. In some way it is the closest approach to ours as it provides several possible digits, however fixed in advance, at every step of the recoding and seem to provide more randomness and security than the previous approaches. It is also interesting to remark that both methods can be combined as they rely on different aspects of the exponent recoding.

6 Conclusions

In this work we have proposed a generalization of the traditional fractional wNAF recoding. Our algorithm allows the computation of the representation of an integer using a set of any digits that has 1 in it. We also have given a general formulae to compute the average density of non-zero terms of such representations. We also studied the average density obtained when digits are chosen at random in a given set and suggested a randomized schemes that could be used to provide some additional protection against differential side-channel attacks. It has the advantage of being flexible, as the number of digits and thus the security provided can be chosen at will, and is also efficient as the average density is close to that of the fractional wNAF using the same number of digits.

The main question that arises is that of the possibility to generalize such recoding even further. In other words, is it possible to eliminate the necessity of having 1 in the digit set? We know that not all numbers can be represented using any digit set, so given a digit set $\mathcal{D}$, is there a condition under which any integer can have a $\mathcal{D}$-representation?

References

[1] A. Brauer. On addition chains. *Bulletin of the American Mathematical Society*, 45(10):736–739, 1939.

[2] H. Cohen and G. Frey, editors. *Handbook of Elliptic and Hyperelliptic Cryptography*. Chapman & Hall, 2006.

[3] V. Dimitrov, L. Imbert, and P. K. Mishra. The double-base number system and its application to elliptic curve cryptography. *Mathematics of Computations*, 77, 2008.

[4] Pierre-Alain Fouque, Frédéric Muller, Guillaume Poupard, and Frédéric Valette. Defeating countermeasures based on randomized bsd representations. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004*, volume 3156 of *Lecture Notes in Computer Science*, pages 312–327. Springer Berlin Heidelberg, 2004.

[5] Eleonora Guerrini, Laurent Imbert, and Théo Winterhalter. Randomizing scalar multiplication using exact covering systems of congruences. Cryptology ePrint Archive, Report 2015/475, 2015.

[6] Jae-Cheol Ha and Sang-Jae Moon. Randomized signed-scalar multiplication of ecc to resist power attacks. In *Revised Papers from the 4th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '02, pages 551–563. Springer-Verlag, 2003.

[7] D. Hankerson, A. Menezes, and S. Vanstone. *Guide to Elliptic Curve Cryptography*. Springer, 2004.

[8] M.A. Hasan. Power analysis attacks and algorithmic approaches to their countermeasures for koblitz curve cryptosystems. *Computers, IEEE Transactions on*, 50(10):1071–1083, Oct 2001.

[9] Kouichi Itoh, Jun Yajima, Masahiko Takenaka, and Naoya Torii. Dpa countermeasures by improving the window method. In *Cryptographic Hardware and Embedded Systems - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 303–317. Springer Berlin Heidelberg, 2003.

[10] Tetsuya Izu and Tsuyoshi Takagi. A fast parallel elliptic curve multiplication resistant against side channel attacks. In *Public Key Cryptography*, volume 2274 of *Lecture Notes in Computer Science*, pages 280–296. Springer Berlin Heidelberg, 2002.

[11] Chris Karlof and David Wagner. Hidden markov model cryptanalysis. In *Cryptographic Hardware and Embedded Systems - CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*, pages 17–34. Springer Berlin Heidelberg, 2003.

[12] N. Koblitz. Cm-curves with good cryptographic properties. In *Advances in Cryptology - CRYPTO*, volume 576 of *LNCS*, page 279. Springer, February 1992.

[13] P.-Y. Liardet and N. P. Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. In *Cryptographic Hardware and Embedded Systems - CHES*, pages 391–401. Springer-Verlag, 2001.

[14] B. Möller. Improved techniques for fast exponentiation. In Springer Berlin / Heidelberg, editor, *Information Security and Cryptology — ICISC 2002*, volume 2587 of *LNCS*, pages 298–312, 2003.

[15] Elisabeth Oswald and Manfred Aigner. Randomized addition-subtraction chains as a countermeasure against power attacks. In *Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '01, pages 39–50. Springer-Verlag, 2001.

[16] N. Pippenger. On the evaluation of powers and related problems. In *Foundations of Computer Science, 1976., 17th Annual Symposium on*, pages 258–263, Oct 1976.

[17] E. G. Thurber. On addition chains $l(mn) \leq l(n) - b$ and lower bounds for $c(r)$. *Duke Mathematical Journal*, 40:907–913, 1973.

[18] Colin D. Walter. Breaking the liardet-smart randomized exponentiation algorithm. In *Proceedings of the 5th Conference on Smart Card Research and Advanced Application Conference - Volume 5*, CARDIS'02, pages 7–7, Berkeley, CA, USA, 2002. USENIX Association.

[19] ColinD. Walter. Mist: An efficient, randomized exponentiation algorithm for resisting power analysis. In *Topics in Cryptology — CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 53–66. Springer Berlin Heidelberg, 2002.

[20] Gerald S. Walton. The number of observed classes from a multiple hypergeometric distribution. *Journal of the American Statistical Association*, 81(393):pp. 169–171, 1986.