Random Digit Representation of Integers

Nicolas Méloni, M. Anwar Hasan

Random Digit Representation of Integers

Nicolas Méloni
IMATH
Université de Toulon
Email: nicolas.meloni@univ-tln.fr

M. Anwar Hasan
Department of Electrical
and Computer Engineering
University of Waterloo
Email: ahasan@ece.uwaterloo.ca

***Abstract*—Modular exponentiation, or scalar multiplication, 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 propose a general modular exponentiation algorithm based on a generalization of the frac-wNAF recoding and a new precomputation scheme. We also give general formula for the average density of non-zero therms in these representations, prove that there are infinitely many optimal sets for a given number of digits and show that the asymptotic behavior, when those digits are randomly chosen, is very close to the optimal case.**

I. 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 $2\log_2(k) = 2t$ 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 [18], [15]; see [3], [8] 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 generalize those approaches to any set of digits containing 1. We propose a general recoding algorithm using any digit set containing 1 and give a formulae to compute the average density of non-zero terms of the recoding. In particular, we show that the digit set $\{1, 3, \ldots, 2n-1\}$ used in the fractional wNAF method is optimal among all sets of n digits in terms of average non-zero term density of the recoding, but that there are infinitely many such sets and we prove a simple criteria to determine if a set is optimal or not. We also propose a randomized exponentiation scheme in which the digit set is randomly chosen at the beginning of every exponentiation with a specific precomputation scheme. Surprisingly, we show that the average density of the resulting representations is almost as low as that of optimal recodings, so that the exponentiation computational costs are basically the same.

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, give a formulae to compute its average density of non-zero terms for any set of digits and give a criteria to distinguish optimal digit sets. In Section 4 we describe our new randomized exponentiation scheme, study its average density, propose a specific method for the precomputation phase of the exponentiation and compare the overall cost of our method to that of the fractional wNAF. In Section 5 we discuss the security provided by the new scheme and propose some comparisons with previous related works.

II. Preliminaries

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 $\mathcal{D}$ of positive integers.

Then compute g^k using Algorithm 1. From this perspective, most successive improvements of the `square-and-multiply` algorithm can be viewed as new recoding schemes using different sets of digits and providing sparser and sparser recoding, that is with as few non-zero terms as possible. For instance, the computations $h \leftarrow h^2 \times h^{2k}$ and $h \leftarrow (h \times h^k)^2$ being equivalent, it is possible to restrict $\mathcal{D}$ to odd integers. When inversion is cheap, one can extend the recoding scheme using the set $\mathcal{D}$ to negative digits, thus considering the larger set $\overline{\mathcal{D}} = \mathcal{D} \cup \{-d_1, \ldots, -d_l\}$.

Finally it was shown that one is not bound to consider the 2^w-ary representation of k for a fixed w.

All put together, the various improvements give the *fractional windows* or *frac w-NAF* proposed by Möller [15]. In that case, $\mathcal{D} = \{1, 3, \ldots, 2n-1\}$. Möller proved that the average density of non-zero terms of this representation is $\frac{1}{a+1}$, where $a = W_n + \frac{2n}{2^{W_n}}$ and $W_n = \lfloor \log_2 2n - 1 \rfloor$. 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 one to consider ternary or hybrid binary/ternary representations [4] and fast group endomorphism have been used in producing complex representations such as the τNAF on Koblitz curves [13].*

III. Random digit representation (RDR)

Let $\mathcal{D} = \{d_1, \ldots, d_l\}$ be a set of odd integers. Let $\overline{\mathcal{D}} = \mathcal{D} \cup \{-d_1, \ldots, -d_l\}$. 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 necessarily hold. Indeed, with $\mathcal{D} = \{5, 13\}$, 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 $\{1, d_2, \ldots, d_n\}$.

A. 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 $W_n = \lfloor \log_2(\max_i(d_i)) \rfloor$.

In order to define the recoding map, we first need to define, for all odd integers k, $w_{max}(k)$ as the largest integer $w \leq W_n + 2$ such that there exists a digit $d_i \in \mathcal{D}$ satisfying the following two conditions:

1) $d_i < k$,
2) $p_w(k) \in \overline{\mathcal{D}}_w$.

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

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

It is worth mentioning that the map is well defined, that is to say that $digit_{\mathcal{D}}(k)$ exists for all odd k. Indeed $1 \in \mathcal{D}$ which implies that $1 \in \mathcal{D}_w$ for any w so that $W_{max} \geq 2$. 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 $\mathcal{D} = \{1, d_2, \ldots, d_n\}$
Ensure: $k = (k_t k_{t-1} \ldots k_1 k_0)_2, k_i \in \overline{\mathcal{D}} \cup \{0\}$
1: $i = 0$
2: **while** $k \neq 0$ **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 $\mathcal{D} = \{1, 3, \ldots, 2n-1\}$ we obtain exactly the* fractional windows *recoding.*

Example 1. *Let $k = 31415$ and $\mathcal{D} = \{1, 3, 23, 27\}$. We have $W_n = 4$, $\mathcal{D}_2 = \{1, 3\}$, $\mathcal{D}_3 = \{1, 3, 7\}$, $\mathcal{D}_4 = \{1, 3, 7, 11\}$ 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 $k \leftarrow \frac{k-23}{2} = 15696$.*
3) *$k_1 = k_2 = k_3 = k_4 = 0$ and $k \leftarrow \frac{k}{2^4} = 981$.*

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* $W_{\max} = 4$ *and* $digit_{\mathcal{D}}(k) = -27$ *as* $p_4(27) = 11$*,*
5) $k_5 = -27$ *and* $k \leftarrow \frac{k+27}{2} = 504$*,*
6) $k_6 = k_7 = k_8 = 0$ *and* $k \leftarrow k/8 = 63$*,*
7) $k \mod 2^6 \equiv -1$ *so* $W_{max} = 6$*,* $k_9 = -1$ *and* $k \leftarrow \frac{k+1}{2}$
8) $k_{10} = k_{11} = k_{12} = k_{13} = k_{14} = 0$*,* $k \leftarrow \frac{k}{2^5}$
9) $k = k_{15} = 1$.

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

B. Average density

Theorem 1. *Let* k *be an integer and* $\mathcal{D} = \{1, d_2, \ldots, d_n\}$ *a set of digits. For all* $w \geq 2$ *we define* $D(w) = \frac{\#\overline{\mathcal{D}}_w}{2^{w-1}}$*. Then the asymptotic average density of non-zero terms achieved by the random digit representation is* $\frac{1}{a_{\mathcal{D}}+1}$*, where*

$$a_{\mathcal{D}} = 2D(W_n + 2) + \sum_{w=2}^{W_n+1} D(w).$$

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 $w_{max}(k) = W_{max} = w$ for every $w \leq W_n + 2$. As given in Theorem 1, $D(w)$ is the probability that the residue of a given odd integer modulo 2^w lies in $\overline{\mathcal{D}}_w$. By construction we have that $P(W_n+2) = D(W_n+2)$. For lower values of w, we must evaluate the probability that a residue is in $\overline{\mathcal{D}}_w$ but not in $\overline{\mathcal{D}}_{w+1}$. It is clear that $(d \mod 2^{w+1} \in \overline{\mathcal{D}}_{w+1}) \Rightarrow (d \mod 2^w \in \overline{\mathcal{D}}_w)$ which implies that, for $w < W_n+2$, $P(w) = D(w)-D(w+1)$.

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

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

If $W_{max} < W_n + 2$, 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 $a_{\mathcal{D}}$, the average number of zeros following a non-zero digit in the RDR of an integer in terms of $D(w)$:

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

□

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

$$a_{\mathcal{D}} = W_n + \frac{2n}{2^{W_n}},$$

which corresponds to the standard result on the average density of the frac-w*NAF 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*

$$a_{\mathcal{D}} = 2 \times \frac{1}{4} + \frac{1}{2} + 1 + 1 + 1 = 4.$$

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.*

C. Optimal digit sets

The random digit representation obtained from Algorithm 2 is a generalization of the frac-wNAF recoding. Now we show that, for a given number of digits n, the lowest asymptotic density of non-zero terms is that of the corresponding frac-wNAF recoding but can be obtained with infinitely many digit sets.

Definition 1. *Let* $n > 0$ *and* $\mathbb{D}_n$ *be the set of all sets of odd integers of the form* $\{1, d_2, \ldots, d_n\}$*. Then* $\mathcal{D} \in \mathbb{D}_n$ *is an* ***optimal digit set*** *if*

$$a_{\mathcal{D}} = \max_{\mathcal{D}' \in \mathbb{D}_n} (a_{\mathcal{D}'}).$$

Theorem 2. *Let* $n > 0$*,* $w_n = \lfloor \log_2 n \rfloor$ *and* $\mathcal{D} \in \mathbb{D}_n$*. If* $\mathcal{D}$ *is an optimal digit then*

$$a_{\mathcal{D}} = w_n + \frac{n}{2^{w_n}} + 1.$$

Proof. From Theorem 1 we know that

$$a_{\mathcal{D}} = \underbrace{2D(W_n+2)}_{A} + \underbrace{\sum_{i=w_n+3}^{W_n+1} D(i)}_{B} + \underbrace{\sum_{i=2}^{w_n+2} D(i)}_{C}.$$

First we remark that $\forall w \geq 2, \#\overline{\mathcal{D}}_w \leq \min(2^{w-1}, 2n)$. By definition of w_n, we have $2n < 2^{w_n+2}$, thus $\#\overline{\mathcal{D}}_w \leq 2n$ for all $w \geq w_n + 3$.

So we have

$$A = 2\frac{2n}{2^{W_n+1}} = \frac{n}{2^{W_n-1}},$$

$$\begin{aligned} B &\leq \sum_{i=w_n+3}^{W_n+1} \frac{2n}{2^{i-1}} \\ &\leq \frac{4n}{2^{w_n+2}} \times (1 - \frac{1}{2^{W_n-w_n-1}}) \\ &\leq \frac{n}{2^{w_n}} - \frac{n}{2^{W_n-1}}, \end{aligned}$$

$$C \leq \sum_{i=2}^{w_n+2} 1 \leq w_n + 1.$$

So we have shown that $a_{\mathcal{D}} \leq w_n + \frac{n}{2^{w_n}} + 1$. To conclude we just need to remark that the inequality becomes an equality for $\mathcal{D} = \{1, 3, \ldots, 2n-1\}$. Indeed if n is a power of two then $n = 2^{w_n}$, $W_n = w_n$ and $a_{\mathcal{D}} = w_n + 2 = w_n + \frac{n}{2^{w_n}} + 1$. If n is not a power of two then $W_n = \lfloor \log_2(2n-1) \rfloor = w_n + 1$ and $2^{W_n} < 2n - 1 < 2^{W_n+1}$ so that $D(2) = \cdots = D(W_n + 1) = 1$ and $2D(W_n + 2) = \frac{4n}{2^{W_n+1}}$, which leads to $a_{\mathcal{D}} = W_n + \frac{n}{2^{W_n-1}} = w_n + 1 + \frac{n}{2^{w_n}}$.

□

Corolary 1. *Let $n > 0$, $w_n = \lfloor \log_2 n \rfloor$ and $\mathcal{D} \in \mathbb{D}_n$. Then $\mathcal{D}$ is an optimal digit set if and only if:*

$$\#\overline{\mathcal{D}}_{w_n+3} = 2n \text{ and } \#\overline{\mathcal{D}}_{w_n+2} = 1.$$

Proof. From the proof of Theorem 2, we know that $\mathcal{D}$ is optimal if and only if for all $i \geq w_n + 3, \#\overline{\mathcal{D}}_i = 2n$ and for all $2 \leq i \leq w_n + 2, D(i) = 1$ i.e. $\#\overline{\mathcal{D}}_i = 2^{i-1}$. To prove this corollary it is enough to prove that, on the one hand, $\#\overline{\mathcal{D}}_{w_n+3} = 2n \Rightarrow \#\overline{\mathcal{D}}_w = 2n$ for all $w \geq w_n + 3$ and that, on the other hand, for all $2 < i$, $(D(i) = 1) \Rightarrow (D(i-1) = 1)$.

Both properties result from the fact that $\forall w \geq 2$, $\#\overline{\mathcal{D}}_{w+1} \leq 2\left(\#\overline{\mathcal{D}}_w\right)$. Indeed, any $d \in \#\overline{\mathcal{D}}_{w+1}$ can be written as $d' + b2^w$ with $b \in \{0, 1\}$ and $d' \in \overline{\mathcal{D}}_w$ so that there are at most twice as many elements in $\overline{\mathcal{D}}_{w+1}$ than in $\overline{\mathcal{D}}_w$. It implies that $D(w+1) = \frac{\#\overline{\mathcal{D}}_{w+1}}{2^w} \leq 2\frac{\#\overline{\mathcal{D}}_w}{2^w} \leq D(w)$ which in turn implies that $(D(w) = 1) \Rightarrow (D(w-1) = 1)$. Also, $\forall w \leq w_n + 3$, $2n \leq \#\overline{\mathcal{D}}_{w_n+3} \leq \#\overline{\mathcal{D}}_w \leq \min(2n, 2^w) \leq 2n$, which concludes the proof. □

Example 4. *Using the previous corollary it is easy to show that the previously seen set $\mathcal{D} = \{1, 3, 23, 27\}$ is optimal. We have $n = 4$ and $w_n = 2$, so we just need to check that $\#\overline{\mathcal{D}}_4 = \#\{1, 3, 5, 7, 9, 11, 13, 15\} = 2^3$ and $\#\overline{\mathcal{D}}_5 = \#\{1, 3, 5, 9, 23, 27, 29, 31\} = 2n$.*

Remark 3. *Corollary 1 directly implies that, for all n, there are infinitely many optimal digit sets. For instance, all sets of the form*

$$\mathcal{D} = \{1, 3, \ldots, 2n-3\} \cup \{2n - 1 + 2^w\}$$

with $w \geq w_n + 3$ are optimal. More generally, for a given number n, all sets of the form

$$\{1, 3 + t_2 2^{w_n+3}, \ldots, 2n - 1 + t_n 2^{w_n+3}\}$$

with $(t_2, \ldots, t_n) \in \mathbb{N}^{n-1}$ are optimal.

IV. Randomized exponentiation scheme

Algorithm 2 allows us to compute the random digit representation (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 l two integers satisfying $l \leq (m+1)/2$. One can compute g^k using the following scheme:

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

One obvious advantage of such a scheme is that it provides an added resistance to differential power analysis alongside with other schemes such as point blinding. Indeed, from one exponentiation to the other, it ensures that the operation flow will be completely different.

A. 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, let $N = \frac{m+1}{2}$ and $W_N = \lfloor \log_2 N \rfloor$, we note that by definition all $d \in \mathcal{D}$ are smaller than 2^{W_N+1}, which implies that $(2^{W_N+2} - d) \neq d$ and thus $\overline{\mathcal{D}_{W_N+2}} = 2l$. Evaluating $D(w)$ for smaller value of w becomes a harder problem. If an integer d is in $\mathcal{D}$, then all integers 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 $w \leq W_N + 1$ by:

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

A simple way to consider this problem is to see it as an urn problem. Let us consider N balls corresponding to each integer of 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 $\mathcal{R}_w$, each pair $(i \mod 2^w, 2^w - (i \mod 2^w))$ for i in $\{1, 3, \ldots, 2^{w-1} - 1\}$ being a representative of one of them. Finally, define E_w^i as the number of representatives of each of those classes. Our problem consists of 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 [21] 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 I. 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.

m	$\#\mathcal{D}$	RDR	wNAF
7	2	3.833	4
15	4	4.771	5
31	8	5.728	6
63	16	6.706	7
127	32	7.695	8
255	64	8.689	9
511	128	9.686	10
1023	256	10.69	11

TABLE I
INVERSES OF THE DENSITY OF THE RDR AND wNAF USING $\left\lfloor \frac{m+1}{4} \right\rfloor$ DIGITS

B. 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 [17]. It is a very general algorithm that allows 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.

In order to give a clearer view of the computational cost of our generalized recoding using our precomputation scheme we have evaluated the number of group operations and the overall density of non-zero terms of both the RDR and the frac-wNAF in various situations. To do so, for different values of the number of digits n we have:

1) randomly generated up to 1000 random digit sets of size n,
2) randomly generated up to 1000 random optimal digit sets of size n,
3) randomly generated 1000 1024-bit integers,
4) computed the inverse of the average density $a_{\mathcal{D}}+1$ and the number of group operations of the RDR for each digit set in terms of squarings (S) and multiplications (M)
5) done the same for the frac-wNAF using digit set $\mathcal{D} = \{1, 3, \ldots, 2n-1\}$.

The results of our experiments are summarized in Table II. It clearly shows that the overhead cost caused by the use of a randomized digit set is negligible compared to the cost of the frac-wNAF recoding. Also, as expected, restricting the RDR to optimal digit sets leads to even faster exponentiations with recodings with the same density of non-zero terms as those obtained with the frac-wNAF method.

Digit set size	Method	$a_{\mathcal{D}}+1$	group operation count
8	RDR	5.701	1023S+191M
	Opt. RDR	5.970	1024S+183M
	frac-wNAF	5.997	1023S+178M
16	RDR	6.666	1023S+175M
	Opt. RDR	6.960	1023S+169M
	frac-wNAF	6.962	1022S+161M
24	RDR	7.209	1023S+175M
	Opt. RDR	7.454	1023S+167M
	frac-wNAF	7.454	1023S+160M
32	RDR	7.634	1023S+175M
	Opt. RDR	7.940	1023S+170M
	frac-wNAF	7.950	1022S+160M
48	RDR	8.178	1023S+190M
	Opt. RDR	8.434	1023S+180M
	frac-wNAF	8.440	1022S+168M
64	RDR	8.692	1023S+207M
	Opt. RDR	8.922	1023S+196M
	frac-wNAF	8.940	1022S+177M

TABLE II
DENSITY AND GROUP OPERATION COUNTS OF THE RANDOM DIGIT REPRESENTATION AND FRACTIONAL wNAF RECODINGS FOR 1024-BIT EXPONENTS.

V. SIDE-CHANNEL SECURITY

The main interest of randomizing the exponentiation process is to provide an added 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.

A. Differential attacks

Differential power analysis 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 [5] or Liardet-Smart randomized algorithm [19] 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 [12].

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 known 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.

B. Simple power analysis attacks

Simple power analysis 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 [2].

Our algorithm clearly will not have a regular behavior, however that does not signifies that it is vulnerable to simple power analysis 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.

C. Related works

Randomization is a standard way to provide security against differential side-channel attacks. In particular, several randomized exponentiation algorithms have been proposed [7], [16], [11], [20], [9] but the security offered by those methods remains in general uncertain. For example, randomized recodings proposed by Ha and Moon [7] or Oswald and Aigner [16] have been defeated due to little local variation of the data. It was exploited through collision detection [5] or more generally using the hidden Markov model cryptanalysis [12]. In a similar way, some randomization techniques focus on the management of the window in sliding window algorithm [10], [14] and successful attacks have been mounted against some of them [19]. 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 [6]. 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.

VI. 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. Two important results that we have shown are that the optimal density can be achieved by infinitely many sets and that there is a simple criteria to distinguish them. We also studied the average density obtained when digits are chosen at random from a given set. The surprising result is that recodings obtained from random digit sets almost have the same density of non-zero terms as optimal ones. Combined with our new precomputation scheme we have proposed a randomized exponentiation scheme and suggested that it could be used to provide some additional protection against differential power analysis attacks for almost no additional cost.

References

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

[2] Benoît Chevallier-Mames, Mathieu Ciet, and Marc Joye. Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity. *IEEE Transactions on Computers*, 53(6):760–768, 2004.

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

[4] 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.

[5] 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.

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.