Faster Homomorphic Function Evaluation using Non-Integral Base Encoding

Charlotte Bonte[1], Carl Bootland[1], Joppe W. Bos[2], Wouter Castryck[1,3], Ilia Iliashenko[1], and Frederik Vercauteren[1,4]

[1] imec-Cosic, Dept. Electrical Engineering, KU Leuven
[2] NXP Semiconductors
[3] Laboratoire Paul Painlevé, Université de Lille-1
[4] Open Security Research

Abstract. In this paper we present an encoding method for real numbers tailored for homomorphic function evaluation. The choice of the degree of the polynomial modulus used in all popular somewhat homomorphic encryption schemes is dominated by security considerations, while with the current encoding techniques the correctness requirement allows for much smaller values. We introduce a generic encoding method using expansions with respect to a non-integral base, which exploits this large degree at the benefit of reducing the growth of the coefficients when performing homomorphic operations. This allows one to choose a smaller plaintext coefficient modulus which results in a significant reduction of the running time. We illustrate our approach by applying this encoding in the setting of homomorphic electricity load forecasting for the smart grid which results in a speed-up by a factor 13 compared to previous work, where encoding was done using balanced ternary expansions.

1 Introduction

The cryptographic technique which allows an untrusted entity to perform arbitrary computation on encrypted data is known as fully homomorphic encryption. The first such construction was based on ideal lattices and was presented by Gentry in 2009 [24]. When the algorithm applied to the encrypted data is known in advance one can use a *somewhat homomorphic encryption* (SHE) scheme which only allows to perform a limited number of computational steps on the encrypted data. Such schemes are significantly more efficient in practice.

In all popular SHE schemes, the plaintext space is a ring of the form $R_t = \mathbb{Z}_t[X]/(f(X))$, where $t \geq 2$ is a small integer called the coefficient modulus, and $f(X) \in \mathbb{Z}[X]$ is a monic irreducible degree d polynomial called the polynomial modulus. Usually one lets $f(X)$ be a cyclotomic polynomial, where for reasons of

This work was supported by the European Commission under the ICT programme with contract H2020-ICT-2014-1 644209 HEAT, and through the European Research Council under the FP7/2007-2013 programme with ERC Grant Agreement 615722 MOTMELSUM. The second author is also supported by a PhD fellowship of the Research Foundation - Flanders (FWO).

performance the most popular choices are the power-of-two cyclotomics $X^d + 1$ where $d = 2^k$ for some positive integer k, which are maximally sparse. In this case arithmetic in R_t can be performed efficiently using the fast Fourier transform, which is used in many lattice-based constructions (e.g. [8,9,10,34]) and most implementations (e.g. [3,6,7,25,26,29,32]).

One interesting problem relates to the *encoding* of the input data of the algorithm such that it can be represented as elements of R_t and such that one obtains a meaningful outcome after the encrypted result is decrypted and decoded. This means that addition and multiplication of the input data must agree with the corresponding operations in R_t up to the depth of the envisaged SHE computation. An active research area investigates different such encoding techniques, which are often application-specific and dependent on the type of the input data. For the sake of exposition we will concentrate on the particularly interesting and popular setting where the input data consists of finite precision real numbers θ, even though our discussion below is fairly generic. The main idea, going back to Dowlin et al. [19] (see also [20,27,31]) and analyzed in more detail by Costache et al. [16], is to expand θ with respect to a base b

$$\theta = a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s} \quad (1)$$

using integer digits a_i, after which one replaces b by X to end up inside the Laurent polynomial ring $\mathbb{Z}[X^{\pm 1}]$. One then reduces the digits a_i modulo t and applies the ring homomorphism to R_t defined by

$$\iota : \mathbb{Z}_t[X^{\pm 1}] \to R_t : \begin{cases} X & \mapsto X, \\ X^{-1} & \mapsto -g(X) \cdot f(0)^{-1}, \end{cases}$$

where we write $f(X) = Xg(X) + f(0)$ and it is assumed that $f(0)$ is invertible modulo t; this is always true for cyclotomic polynomials, or for factors of them. The quantity $r + s$ will sometimes be referred to as the *degree* of the encoding (where we assume that $a_r a_{-s} \neq 0$). For power-of-two-cyclotomics the homomorphism ι amounts to letting $X^{-1} \mapsto -X^{d-1}$, so that the encoding of (1) is given by[5] $a_r X^r + a_{r-1} X^{r-1} + \cdots + a_1 X + a_0 - a_{-1} X^{d-1} - a_{-2} X^{d-2} - \cdots - a_{-s} X^{d-s}$.

Decoding is done through the inverse of the restriction $\iota|_{\mathbb{Z}_t[X^{\pm 1}]_{[-\ell,m]}}$ where

$$\mathbb{Z}_t[X^{\pm 1}]_{[-\ell,m]} = \{\, a_m X^m + a_{m-1} X^{m-1} + \ldots + a_{-\ell} X^{-\ell} \mid a_i \in \mathbb{Z}_t \text{ for all } i \,\}$$

is a subset of Laurent polynomials whose monomials have bounded exponents. If $\ell + m + 1 = d$ then this restriction of ι is indeed invertible as a $\mathbb{Z}_t$-linear map. The precise choice of ℓ, m depends on the data encoded. After applying this inverse, one replaces the coefficients by their representants in $\{-\lfloor (t-1)/2 \rfloor, \ldots, \lceil (t-1)/2 \rceil\}$ to end up with an expression in $\mathbb{Z}[X^{\pm 1}]$, and evaluates the result at $X = b$. Ensuring that decoding is correct to a given computational depth places constraints on the parameters t and d, in order to avoid ending up outside the box depicted in Figure 1 if the computation were to be carried out directly in

[5] In fact in [16] it is mentioned that inverting X is only possible in the power-of-two cyclotomic case, but this seems to be overcareful.

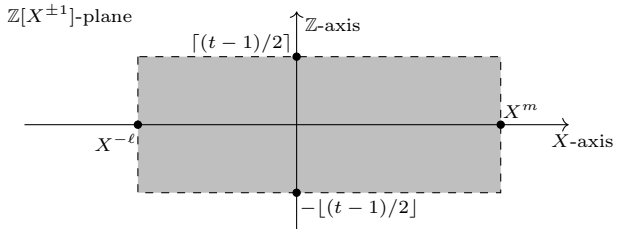

Fig. 1. Box in which to stay during computation, where $\ell + m + 1 = d$.

$\mathbb{Z}[X^{\pm 1}]$. In terms of R_t we will often refer to this event as the 'wrapping around' of the encoded data modulo t or $f(X)$, although we note that this is an abuse of language. In the case of power-of-two cyclotomics, ending up above or below the box does indeed correspond to wrapping around modulo t, but ending up at the left or the right of the box corresponds to a mix-up of the high degree terms and the low degree terms.

The precise constraints on t and d not only depend on the complexity of the computation, but also on the type of expansion (1) used in the encoding. Dowlin et al. suggest to use balanced b-ary expansions with respect to an odd base $b \in \mathbb{Z}_{\geq 3}$, which means that the digits are taken from $\{-(b-1)/2, \ldots, (b-1)/2\}$. Such expansions have been used for centuries going back at least to Colson (1726) and Cauchy (1840) in the quest for more efficient arithmetic.

If we fix a precision, then for smaller b the balanced b-ary expansions are longer but the coefficients are smaller, this implies the need for a larger d but smaller t. Similarly for larger bases the expansions become shorter but have larger coefficients leading to smaller d but larger t. For the application to somewhat homomorphic encryption considered in [6,16] the security requirements ask for a very large d, so that the best choice is to use as small a base as possible, namely $b = 3$, with digits in $\{\pm 1, 0\}$. Even for this smallest choice the resulting lower bound on t is very large and the bound on d is much smaller than that coming from the cryptographic requirements. To illustrate this, we recall the concrete figures from the paper [6], which uses the Fan-Vercauteren (FV) somewhat homomorphic encryption scheme [23] for privacy-friendly prediction of electricity consumption in the setting of the smart grid. Here the authors use $d = 4096$ for cryptographic reasons, which is an optimistic choice that leads to 80-bit security only (and maybe even a few bits less than that [1]). On the other hand using balanced ternary expansions, correct decoding is guaranteed as soon as $d \geq 368$, which is even a conservative estimate. This eventually leads to the huge bound $t \gtrapprox 2^{107}$, which is overcome by decomposing R_t into 13 factors using the Chinese Remainder Theorem (CRT). This is then used to homomorphically forecast the electricity usage for the next half hour for a small apartment complex of 10 households in about half a minute, using a sequential implementation.

The discrepancy between the requirements coming from correct decoding and those coming from security considerations suggests that other possible expan-

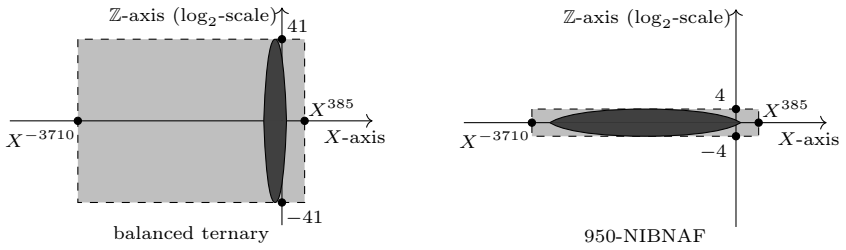

Fig. 2. Comparison of the amount of plaintext space which is actually used in the setting of [6], where $d = 4096$. More precise figures to be found in Section 4.

sions may be better suited for use with SHE. In this paper we introduce a generic encoding technique, using very sparse expansions having digits in $\{\pm 1, 0\}$ with respect to a *non-integral* base $b_w > 1$, where w is a sparseness measure. These expansions will be said to be of 'non-integral base non-adjacent form' with window size w, abbreviated to w-NIBNAF. Increasing w makes the degrees of the resulting Laurent polynomial encodings grow and decreases the growth of the coefficients when performing operations; hence lowering the bound on t. Our encoding technique is especially useful when using finite precision real numbers, but could also serve in dealing with finite precision complex numbers or even with integers, despite the fact that b_w is non-integral (this would require a careful precision analysis which is avoided here).

We demonstrate that this technique results in significant performance increases by re-doing the experiments from [6]. Along with a more careful precision analysis which is tailored for this specific use case, using 950-NIBNAF expansions we end up with the dramatically reduced bound $t \geq 33$. It is not entirely honest to compare this to $t \gtrapprox 2^{107}$ because of our better precision analysis; as explained in Section 4 it makes more sense to compare the new bound to $t \gtrapprox 2^{42}$, but the reduction remains huge. As the reader can see in Figure 2 this is explained by the fact that the data is spread more evenly across the plaintext space during computation. As a consequence we avoid the need for CRT decomposition and thus reduce the running time by a factor 13, showing that the same homomorphic forecasting can be done in only 2.5 seconds.

Remark. An alternative recent proposal for encoding using a non-integral base can be found in [15], which targets efficient evaluation of the discrete Fourier transform on encrypted data. Here the authors work exclusively in the power-of-two cyclotomic setting $f(X) = X^d + 1$, and the input data consists of complex numbers θ which are expanded with respect to the base $b = \zeta$, where ζ is a primitive $2d$-th root of unity, i.e. a root of $f(X)$; a similar idea was used in [12]. One nice feature of this approach is that the correctness of decoding is not affected by wrapping around modulo $f(X)$. To find a sparse expansion they use the LLL algorithm [28], but for arbitrary complex inputs the digits become rather large when compared to w-NIBNAF.

2 Encoding data using w-NIBNAF

Our approach in reducing the lower bound on the plaintext modulus t is to use encodings for which many of the coefficients are zero. In this respect, a first improvement over balanced ternary expansions is obtained by using the non-adjacent form (NAF) representations which were introduced by Reitweisner in 1960 for speeding up early multiplication algorithms [33]. We note that independent work by Cheon et al. [11] also mentions the advantages of using NAF encodings.

Definition 1. *The* non-adjacent form (NAF) representation *of a real number θ is an expansion of θ to the base $b = 2$ with coefficients in $\{-1, 0, 1\}$ such that any two adjacent coefficients are not both non-zero.*

The NAF representation has been generalized [13]: for an integer $w \geq 1$ (called the 'window size') one can ensure that in any window of w consecutive coefficients at most one of them is non-zero. This is possible to base $b = 2$ but for $w > 2$ one requires larger coefficients.

Definition 2. *Let $w \geq 1$ be an integer. A* w-NAF representation *of a real number θ is an expansion of θ with base 2 and whose non-zero coefficients are odd and less than 2^{w-1} in absolute value such that for every set of w consecutive coefficients at most one of them is non-zero.*

We see that NAF is just the special case of w-NAF for $w = 2$. Unfortunately, due to the fact that the coefficients are taken from a much larger set, using w-NAF encodings in the SHE setting actually gives larger bounds on both t and d for increasing w. Therefore this is not useful for our purposes.

Ideally, we want the coefficients in our expansions to be members of $\{\pm 1, 0\}$ with many equal to 0, as this leads to the slowest growth in coefficient sizes, allowing us to use smaller values for t. This would come at the expense of using longer encodings, but remember that we have a lot of manoeuvring space on the d side. One way to achieve this goal is to use a *non-integral* base $b > 1$ when computing a non-adjacent form. We first give the definition of a non-integral base non-adjacent form with window size w (w-NIBNAF) representation and then explain where this precise formulation comes from.

Definition 3. *A sequence $a_0, a_1, \ldots, a_n, \ldots$ is a* w-balanced ternary sequence *if it has $a_i \in \{-1, 0, 1\}$ for $i \in \mathbb{Z}_{\geq 0}$ and satisfies the property that each set of w consecutive terms has no more than one non-zero term.*

Definition 4. *Let $\theta \in \mathbb{R}$ and $w \in \mathbb{Z}_{>0}$. Define b_w to be the unique positive real root of the polynomial $F_w(x) = x^{w+1} - x^w - x - 1$. A w-balanced ternary sequence $a_r, a_{r-1}, \ldots, a_1, a_0, a_{-1}, \ldots$ is a w-NIBNAF representation of θ if*

$$\theta = a_r b_w^r + a_{r-1} b_w^{r-1} + \cdots + a_1 b_w + a_0 + a_{-1} b_w^{-1} + \cdots \quad .$$

Below we will show that every $\theta \in \mathbb{R}$ has at least one such w-NIBNAF representation and provide an algorithm to find such a representation. But let us first state a lemma which shows that b_w is well-defined for $w \geq 1$.

Lemma 1. *For an integer $w \geq 1$ the polynomial $F_w(x) = x^{w+1} - x^w - x - 1$ has a unique positive real root $b_w > 1$. The sequence $b_1, b_2, \ldots$ is strictly decreasing and $\lim_{w\to\infty} b_w = 1$. Further, $(x^2+1) \mid F_w(x)$ for $w \equiv 3 \bmod 4$.*

The proof is straightforward and given in Appendix A. The first few values of b_w are as follows

$$\begin{aligned} b_1 &= 1 + \sqrt{2} \approx 2.414214, & b_2 &\approx 1.839287, \\ b_3 &= \tfrac{1}{2}(1+\sqrt{5}) \approx 1.618034, & b_4 &\approx 1.497094, \end{aligned}$$

where we note that b_3 is the golden ratio ϕ.

Since we are using a non-integral base, a w-NIBNAF representation of a fixed-point number has infinitely many non-zero terms in general. To overcome this one approximates the number by terminating the w-NIBNAF representation after some power of the base. We call such a terminated sequence an *approximate w-NIBNAF representation*. There are two straightforward ways of deciding where to terminate: either a fixed power of the base is chosen so that any terms after this are discarded giving an easy bound on the maximal possible error created, or we choose a maximal allowed error in advance and terminate after the first power which gives error less than or equal to this value.

Algorithm 1 below produces for every $\theta \in \mathbb{R}$ a w-NIBNAF representation in the limit as ϵ tends to 0, thereby demonstrating its existence. It takes the form of a greedy algorithm which chooses the closest signed power of the base to θ and then iteratively finds a representation of the difference. Except when θ can be written as $\theta = h(b_w)/b_w^q$, for some polynomial h with coefficients in $\{\pm 1, 0\}$ and $q \in \mathbb{Z}_{\geq 0}$, any w-NIBNAF representation is infinitely long. Hence, we must terminate Algorithm 1 once the iterative input is smaller than some pre-determined precision $\epsilon > 0$.

We now prove that the algorithm works as required.

Lemma 2. *Algorithm 1 produces an approximate w-NIBNAF representation of θ with an error of at most ϵ.*

Proof. Assuming that the algorithm terminates, the output clearly represents θ to within an error of at most size ϵ. First we show that the output is w-NIBNAF. Suppose that the output, on input θ, b_w, ϵ, has at least two non-zero terms, the first being a_d. This implies either that $b_w^d \leq |\theta| < b_w^{d+1}$ and $b_w^{d+1} - |\theta| > |\theta| - b_w^d$ or $b_w^{d-1} < |\theta| \leq b_w^d$ and $b_w^d - |\theta| \leq |\theta| - b_w^{d-1}$. These conditions can be written as $b_w^d \leq |\theta| < \frac{1}{2} b_w^d (1 + b_w)$ and $\frac{1}{2} b_w^{d-1}(1+b_w) \leq |\theta| \leq b_w^d$ respectively, showing that

$$||\theta| - b_w^d| < \max\left\{b_w^d - \tfrac{1}{2} b_w^{d-1}(1+b_w),\ \tfrac{1}{2} b_w^d (1+b_w) - b_w^d\right\} = \tfrac{1}{2} b_w^d (b_w - 1)\ .$$

The algorithm subsequently chooses the closest power of b_w to this smaller value, suppose it is b_w^ℓ. By the same argument with θ replaced by $|\theta| - b_w^d$ we have that

Algorithm 1: `GreedyRepresentation`

Input: θ – the real number to be represented,
b_w – the w-NIBNAF base to be used in the representation,
ϵ – the precision to which the representation is determined.
Output: An approximate w-NIBNAF representation $a_r, a_{r-1}, \ldots$ of θ with error less than ϵ, where $a_i = 0$ if not otherwise specified.

$\sigma \leftarrow \operatorname{sgn}(\theta)$
$t \leftarrow |\theta|$
while $t > \epsilon$ **do**
  $r \leftarrow \lceil \log_{b_w}(t) \rceil$
  if $b_w^r - t > t - b_w^{r-1}$ **then**
    $r \leftarrow r - 1$
  $a_r \leftarrow \sigma$
  $\sigma \leftarrow \sigma \cdot \operatorname{sgn}(t - b_w^r)$
  $t \leftarrow |t - b_w^r|$
Return $(a_i)_i$.

either $b_w^\ell \leq \left||\theta| - b_w^d\right|$ or $\frac{1}{2} b_w^{\ell-1}(1 + b_w) \leq \left||\theta| - b_w^d\right|$ and since b_w^ℓ is larger than $\frac{1}{2} b_w^{\ell-1}(1+b_w)$ the maximal possible value of ℓ, which we denote by $\ell_w(d)$, satisfies

$$\ell_w(d) = \max\left\{ \ell \in \mathbb{Z} \;\middle|\; \tfrac{1}{2} b_w^{\ell-1}(1 + b_w) < \tfrac{1}{2} b_w^d (b_w - 1) \right\} \quad .$$

The condition on ℓ can be rewritten as $b_w^\ell < b_w^{d+1}(b_w - 1)/(b_w + 1)$ which implies that $\ell < d + 1 + \log_{b_w}((b_w - 1)/(b_w + 1))$ and thus

$$\ell_w(d) = d + \left\lceil \log_{b_w}\left(\frac{b_w - 1}{b_w + 1}\right) \right\rceil \quad ,$$

so that the smallest possible difference is independent of d and equal to

$$s(w) := d - \ell_w(d) = -\left\lceil \log_{b_w}\left(\frac{b_w - 1}{b_w + 1}\right) \right\rceil = \left\lfloor \log_{b_w}\left(\frac{b_w + 1}{b_w - 1}\right) \right\rfloor \quad .$$

We thus need to show that $s(w) \geq w$. As w is an integer this is equivalent to

$$\log_{b_w}\left(\frac{b_w + 1}{b_w - 1}\right) \geq w \iff b_w^w \leq \frac{b_w + 1}{b_w - 1} \iff b_w^{w+1} - b_w^w - b_w - 1 \leq 0$$

which holds for all w since $F_w(b_w) = 0$. Note that our algorithm works correctly and deterministically because when $|\theta|$ is exactly half-way between two powers of b_w we pick the larger power. This shows that the output is of the desired form.

Finally, to show that the algorithm terminates we note that the k'th successive difference is bounded above by $\frac{1}{2} b_w^{d-(k-1)s(w)}(b_w - 1)$ and this tends to 0 as k tends to infinity. Therefore after a finite number of steps (at most $\left\lceil \left(d - \log_{b_w}\left(2\epsilon/(b_w - 1)\right)\right)/s(w) \right\rceil + 1$) the difference is smaller than or equal to ϵ and the algorithm terminates. □

The process of encoding works as described in the introduction, i.e. we follow the approach from [16,19] except we use an approximate w-NIBNAF representation instead of the balanced ternary representation. Thus to encode a real number θ we find an approximate w-NIBNAF representation of θ with small enough error and replace each occurrence of b_w by X, after which we apply the map ι to end up in plaintext space R_t. Decoding is almost the same as well, only that after inverting ι and lifting the coefficients to $\mathbb{Z}$ we evaluate the resulting Laurent polynomial at $X = b_w$ rather than $X = 3$, computing the value only to the required precision. Rather than evaluating directly it is best to reduce the Laurent polynomial modulo $F_w(X)$ (or modulo $F_w(X)/(X^2+1)$ if $w \equiv 3 \bmod 4$) so that we only have to compute powers of b_w up to w (respectively $w-2$).

Clearly we can also ask Algorithm 1 to return $\sum_i a_i X^i \in \mathbb{Z}_t[X^{\pm 1}]$, this gives an encoding of θ with maximal error ϵ. Since the input θ of the algorithm can get arbitrarily close to but larger than ϵ, the final term can be $\pm X^h$ where $h = \lfloor \log_{b_w}(2\epsilon/(1+b_w)) \rfloor + 1$. If we are to ensure that the smallest power of the base to appear in any approximate w-NIBNAF representation is b_w^s then we require that if b_w^{s-1} is the nearest power of b_w to the input θ then $|\theta| \leq \epsilon$ so that we must have $\frac{1}{2} b_w^{s-1}(1+b_w) \leq \epsilon$ which implies the smallest precision we can achieve is $\epsilon = b_w^{s-1}(1+b_w)/2$. In particular if we want no negative powers of b_w then the best precision possible using the greedy algorithm is $(1+b_w^{-1})/2 < 1$.

Remark. If one replaces b_w by a smaller base $b > 1$ then Algorithm 1 still produces a w-NIBNAF expansion to precision ϵ: this follows from the proof of Lemma 2. The distinguishing feature of b_w is that it is maximal with respect to this property, so that the resulting expansions become as short as possible.

3 Analysis of coefficient growth during computation

After encoding the input data it is ready for homomorphic computations. This increases both the number of non-zero coefficients as well as the size of these coefficients. Since we are working in the ring R_t there is a risk that our data wraps around modulo t as well as modulo $f(X)$, in the sense explained in the introduction, which we should avoid since this leads to erroneous decoding. Therefore we need to understand the coefficient growth more thoroughly. We simplify the analysis in this section by only considering multiplications and what constraint this puts on t, it is then not hard to generalize this to include additions.

Worst case coefficient growth for w-NIBNAF encodings. Here we analyze the maximal possible size of a coefficient which could occur from computing with w-NIBNAF encodings. Because fresh w-NIBNAF encodings are just approximate w-NIBNAF representations written as elements of R_t we consider finite w-balanced ternary sequences and the multiplication endowed on them from R_t. Equivalently, we consider multiplication in the $\mathbb{Z}[X^{\pm 1}]$-plane depicted in Figure 1. As we ensure in practice that there is no wrap around modulo $f(X)$ this can be ignored in our analysis.

To start the worst case analysis we have the following lower bound; note that the d we use here is *not* that of the degree of $f(X)$.

Lemma 3. *The maximal absolute size of a term that can appear in the product of p arbitrary w-balanced ternary sequences of length $d+1$ is at least*

$$B_w(d,p) := \sum_{k=0}^{\lfloor \lfloor p\lfloor d/w\rfloor/2\rfloor/(\lfloor d/w\rfloor+1)\rfloor} (-1)^k \binom{p}{k}\binom{p-1+\lfloor p\lfloor d/w\rfloor/2\rfloor - k\lfloor d/w\rfloor - k}{p-1} \ .$$

A full proof of this lemma is given in Appendix A but the main idea is to look at the largest coefficient of m^p where m has the maximal number of non-zero coefficients, $\lfloor d/w\rfloor+1$, all being equal to 1 and with exactly $w-1$ zero coefficients between each pair of adjacent non-zero coefficients. The (non-zero) coefficients of m^p are variously known in the literature as extended (or generalized) binomial coefficients or ordinary multinomials; we denote them here by $\binom{p}{k}_n$ defined via

$$\left(1+X+X^2+\ldots+X^{n-1}\right)^p = \sum_{k=0}^{\infty} \binom{p}{k}_n X^k \ ,$$

[22,18,35,21]. In particular the maximal coefficient is the (or a) central one and we can write $B_w(d,p) = \binom{p}{k}_n$ where $k = \lfloor p\lfloor d/w\rfloor/2\rfloor$ and $n = \lfloor d/w\rfloor + 1$.

We note that the w-NIBNAF encoding, using the greedy algorithm with precision $\frac{1}{2}$, of $b_w^{d+w-(d \bmod w)}(b_w-1)/2$ is m so in practice this lower bound is achievable although highly unlikely to occur.

We expect that this lower bound is tight, indeed we were able to prove the following lemma, the proof is also given in Appendix A.

Lemma 4. *Suppose w divides d, then $B_w(d,p)$ equals the maximal absolute size of a term that can be produced by taking the product of p arbitrary w-balanced ternary sequences of length $d+1$.*

We thus make the following conjecture which holds for all small values of p and d we tested and which we assume to be true in general.

Conjecture 1 *The lower bound $B_w(d,p)$ given in Lemma 3 is exact for all d, that is the maximal absolute term size which can occur after multiplying p arbitrary w-balanced ternary sequences of length $d+1$ is $B_w(d,p)$.*

This conjecture seems very plausible since as soon as one multiplicand does not have non-zero coefficients exactly w places apart the non-zero coefficients start to spread out and decrease in value.

To determine $B_w(d,p)$ for fixed p define $n := \lfloor d/w\rfloor + 1$, then we can expand the expression for $B_w(d,p)$ as a 'polynomial' in n of degree $p-1$ where the coefficients depend on the parity of n, see [5] for more details. The first few are:

$$\begin{aligned}
&B_w(d,1) = 1, && B_w(d,2) = n,\\
&B_w(d,3) = \tfrac{1}{8}(6n^2+1) - \tfrac{(-1)^n}{8}, && B_w(d,4) = \tfrac{1}{3}(2n^3+n),\\
&B_w(d,5) = \tfrac{1}{384}(230n^4+70n^2+27) - \tfrac{(-1)^n}{384}(30n^2+27),\\
&B_w(d,6) = \tfrac{1}{20}(11n^5+5n^3+4n).
\end{aligned}$$

Denoting the coefficient of n^{p-1} in these expressions by ℓ_p, it can be shown (see [2] or [5]) that $\lim_{p\to\infty} \sqrt{p}\ell_p = \sqrt{6/\pi}$ and hence we have

$$\lim_{p\to\infty} \log_2(B_w(d,p)) - (p-1)\log_2(n) + \tfrac{1}{2}\log_2\left(\tfrac{\pi p}{6}\right) = 0$$

or equivalently $B_w(d,p) \sim_p \sqrt{6/\pi p}n^{p-1}$. Thus we have the approximation

$$\log_2(B_w(d,p)) \approx (p-1)\log_2(n) - \tfrac{1}{2}\log_2\left(\tfrac{\pi p}{6}\right)$$

which for large enough n (experimentally we found for $n > 1.825\sqrt{p-1/2}$) is an upper bound for $p > 2$. For a guaranteed upper bound we refer to Mattner and Roos [30] where they state, for $n, p \in \mathbb{Z}_{>0}$ with $n \geq 2$, if $p \neq 2$ or $n \in \{2,3,4\}$ then $B_w(d,p) \leq \sqrt{6/(\pi p(n^2-1))}n^p$. This upper bound is in fact a more precise asymptotic limit than that above which only considers the leading coefficient.

Statistical analysis of the coefficient growth. Based on the w-NIBNAF encodings of random numbers in $N \in \left[-2^{40}, 2^{40}\right]$, we try to get an idea of the amount of zero and non-zero coefficients in a fresh encoding without fractional part, obtained by running Algorithm 1 to precision $(1+b_w^{-1})/2$. We also analyze how these proportions change when we perform multiplications. We plot this for different values of w to illustrate the positive effects of using sparser encodings. As a preliminary remark note that the w-NIBNAF encodings produced by Algorithm 1 applied to $-N$ and N are obtained from one another by changing all the signs, so the coefficients -1 and 1 are necessarily distributed evenly.[6]

We know from the definition of a w-NIBNAF expansion that at least $w-1$ among each block of w consecutive coefficients of the expansion will be 0, so we expect for big w that the 0 coefficient occurs a lot more often than ± 1. This is clearly visible in Figure 3. In addition we see an increasing number of 0 coefficients and decreasing number of ± 1 coefficients for increasing w. Thus both the absolute and the relative sparseness of our encodings increase as w increases.

Since the balanced ternary encoding of [16,19] and the 2-NAF encoding [33], only have coefficients in $\{0, \pm 1\}$ it is interesting to compare them to 1-NIBNAF and 2-NIBNAF respectively. We compare them by computing the percentage of zero and non-zero coefficients, in 10 000 encodings of random integers N in $\left[-2^{40}, 2^{40}\right]$. We compute this percentage up to an accuracy of 10^{-2} and consider for our counts all coefficients up to and including the leading coefficient, further zero coefficients are not counted. When we compare the percentages of zero and non-zero coefficients occurring in 1-NIBNAF and balanced ternary in Table 1 we see that for the balanced ternary representation, the occurrences of 0, 1 and -1 coefficients are approximately the same, while for 1-NIBNAF the proportion of 0

[6] This is a desirable property leading to the maximal amount of cancellation during computation. While this does not affect our worst case analysis, in practice where the worst case is extremely unlikely this accounts for a considerable reduction of the size of the coefficient modulus t. If in some application the input encodings happen to be biased towards 1 or -1 then one can work with respect to the *negative* base $-b_w < -1$, by switching the signs of all the digits appearing at an odd index.

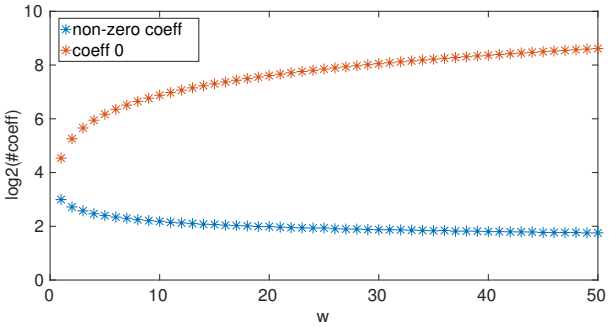

Fig. 3. Plot of $\log_2(\#\text{coeff})$ on the vertical axis against w on the horizontal axis averaged over 10 000 w-NIBNAF encodings of random integers in $\left[-2^{40}, 2^{40}\right]$.

	balanced ternary	1-NIBNAF	2-NAF	2-NIBNAF
zero coefficients	32.25%	48.69%	65.23%	70.46%
non-zero coefficients	67.76%	51.31%	34.77%	29.54%

Table 1. Comparison between the previous encoding techniques and w-NIBNAF

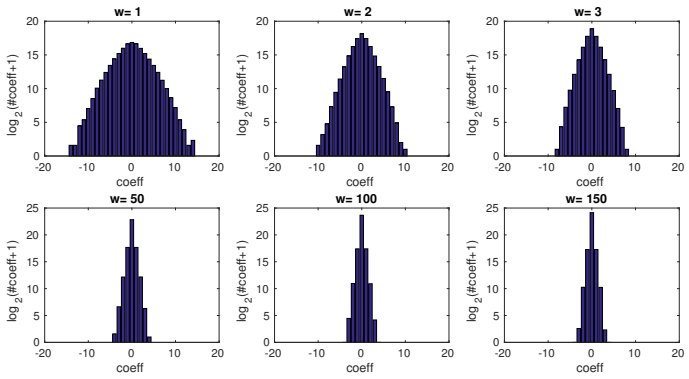

Fig. 4. Plot of $\log_2(\#\text{coeff}+1)$ on the vertical axis against the respective value of the coefficient on the horizontal axis for the result of 10 000 multiplications of two w-NIBNAF encodings of random numbers between $\left[-2^{40}, 2^{40}\right]$.

coefficients is larger than that of 1 or -1. Hence we can conclude that 1-NIBNAF encodings will be sparser than the balanced ternary encodings even though the window size is the same. For 2-NIBNAF we also see an improvement in terms of sparseness of the encoding compared to 2-NAF.

The next step is to investigate what happens to the coefficients when we multiply two encodings. From Figure 4 we see that when w increases the max-

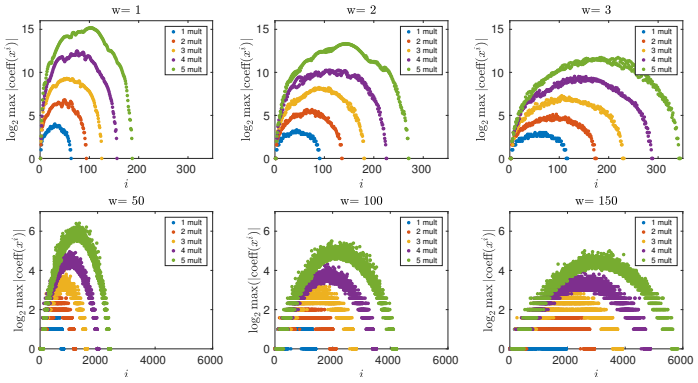

Fig. 5. $\log_2$ of the maximum absolute value of the coefficient of x^i seen during 10 000 products of two w-NIBNAF encodings of random numbers in $\left[-2^{40}, 2^{40}\right]$ against i.

imal size of the resulting coefficients becomes smaller. So the plots confirm the expected result that sparser encodings lead to a reduction in the size of the resulting coefficients after one multiplication. Next, we investigate the behaviour for an increasing amount of multiplications. In Figure 5 one observes that for a fixed number of multiplications the maximum coefficient, considering all coefficients in the resulting polynomial, decreases as w increases and the maximum degree of the polynomial increases as w increases. This confirms that increasing the degree of the polynomial, in order to make it more sparse, has the desirable effect of decreasing the size of the coefficients. Figure 5 also shows that based on the result of one multiplication we can even estimate the maximum value of the average coefficients of x^i for a specific number of multiplications by scaling the result for one multiplication.

To summarize, we plot the number of bits of the maximum coefficient of the polynomial that is the result of a certain fixed amount of multiplications as a function of w in Figure 6. From this figure we clearly see that the maximal coefficient decreases when w increases and hence the original encoding polynomial is sparser. In addition we see that the effect of the sparseness of the encoding on the size of the resulting maximal coefficient is bigger when the amount of multiplications increases. However the gain of sparser encodings decreases as w becomes bigger. Furthermore, Figure 6 shows that the bound given in Lemma 3 is much bigger than the observed upper bound we get from 10 000 samples.

4 Practical impact

We encounter the following constraints on the plaintext coefficient modulus t while homomorphically computing with polynomial encodings of finite precision real numbers. The first constraint comes from the correctness requirement of the SHE scheme: the noise inside the ciphertext should not exceed a certain level during the computations, otherwise decryption fails. Since an increase of the

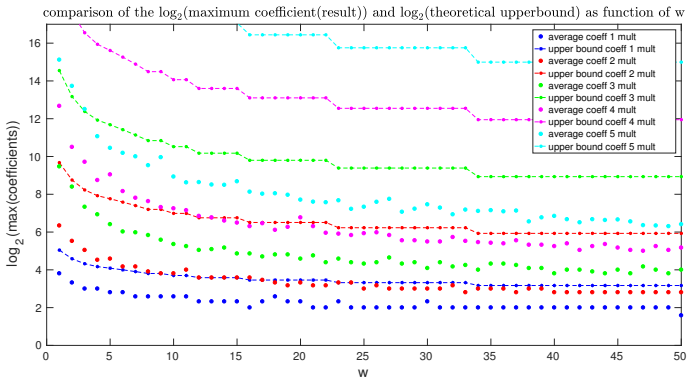

Fig. 6. $\log_2$ of the observed and theoretical maximum absolute coefficient of the result of multiplying w-NIBNAF encodings of random numbers in $\left[-2^{40}, 2^{40}\right]$ against w.

plaintext modulus expands the noise this places an upper bound on the possible t which can be used. The second constraint does not relate to SHE but to the circuit itself. After any arithmetic operation the polynomial coefficients tend to grow. Given that fact, one should take a big enough plaintext modulus in order to prevent or mitigate possible wrapping around modulo t. This determines a lower bound on the range of possible values of t. In practice, for deep enough circuits these two constraints are incompatible, i.e. there is no interval from which t can be chosen. However, the plaintext space R_t can be split into smaller rings $R_{t_1}, \ldots, R_{t_k}$ with $t = \prod_{i=1}^{k} t_i$ using the Chinese Remainder Theorem (CRT). This technique [8] allows us to take the modulus big enough for correct evaluation of the circuit and then perform k threads of the homomorphic algorithm over $\{R_{t_i}\}_i$. These k output polynomials will then be combined into the final output, again by CRT. This approach needs k times more memory and time than the case of a single modulus. Thus the problem is mostly about reducing the number of factors of t needed.

An a priori lower bound on t can be derived using the worst case scenario in which the final output has the maximal possible coefficient, which was analyzed in Section 3. If we use w-NIBNAF encodings for increasing values of w then this lower bound will decrease, eventually leading to fewer CRT factors; here a concern is not to take w too large to prevent wrapping around modulo $f(X)$. In practice though, we can take t considerably smaller because the worst case occurs with a negligible probability, which even decreases for circuits having a bigger multiplicative depth. Moreover, we can allow the least significant coefficients of the fractional part to wrap around modulo t with no harm to the final results.

In this section we revisit the homomorphic method for electricity load forecasting described in [6] and demonstrate that by using w-NIBNAF encodings, by ignoring the unlikely worst cases, and by tolerating minor precision losses we can reduce the number of CRT factors from $k = 13$ to $k = 1$, thereby enhancing its practical performance by a factor 13. We recall that [6] uses the Fan-Vercauteren

SHE scheme [23], along with the group method of data handling (GMDH) as a prediction tool; we refer to [6, §3] for a quick introduction to this method. Due to the fact that 80 percent of electricity meter devices in the European Union should be replaced with smart meters by 2020, this application may mitigate some emerging privacy and efficiency issues.

Experimental setup. For comparison's sake we mimic the treatment in [6] as closely as possible. In particular we also use the real world measurements obtained from the smart meter electricity trials performed in Ireland [14]. This dataset [14] contains observed electricity consumption of over 5000 residential and commercial buildings during 30 minute intervals. We use aggregated consumption data of 10 buildings. Given previous consumption data with some additional information, the GMDH network has the goal of predicting electricity demand for the next time period. Concretely, it requires 51 input parameters: the 48 previous measurements plus the day of the week, the month and the temperature. There are three hidden layers with 8, 4, 2 nodes, respectively. A single output node provides the electricity consumption prediction for the next half hour. Recall that a node is just a bivariate quadratic polynomial evaluation.

The plaintext space is of the form $R_t = \mathbb{Z}_t[X]/(X^{4096}+1)$, where the degree $d = 4096$ is motivated by the security level of 80 bits which is targetted in [6]; recent work by Albrecht [1] implies that the actual level of security is slightly less than that. Inside R_t the terms corresponding to the fractional parts and those corresponding to the integral parts come closer together after each multiplication. Wrapping around modulo $X^{4096}+1$, i.e. ending up at the left or at the right of the box depicted in Figure 1, means that inside R_t these integer and fractional parts start to overlap. In this case it is no longer possible to decode correctly. We encode the input data using approximate w-NIBNAF representations with a fixed number of integer and fractional digits. When increasing the window size w one should take into account that the precision of the corresponding encodings changes as well. To maintain the same accuracy of the algorithm it is important to keep the precision fixed, hence for bigger w's the smaller base b_w should result in an increase of the number of coefficients used by an encoding. Starting with the balanced ternary expansion (BTE), for any $w > 2$, the numbers $\ell(w)_i$ and $\ell(w)_f$ of integer and fractional digits should be expanded according to $\ell(w)_i = (\ell(\mathrm{BTE})_i - 1) \cdot \log_{b_w} 3 + 1$, $\ell(w)_f = -\lfloor \log_{b_w} e_f \rfloor$, where e_f is the maximal error of an approximate w-NIBNAF representation such that the prediction algorithm preserves the same accuracy. Empirically we found that the GMDH network demonstrates reasonable absolute and relative errors when $\ell(\mathrm{BTE})_i^{\mathrm{inp}} = 4$ and $e_f^{\mathrm{inp}} = 1$ for the input and $\ell(\mathrm{BTE})_i^{\mathrm{pol}} = 2$ and $e_f^{\mathrm{pol}} = 0.02032$ for the coefficients of the nodes (quadratic polynomials).

Results. The results reported in this section are obtained running the same software and hardware as in [6]: namely, FV-NFLlib software library [17] running on a laptop equipped with an Intel Core i5-3427U CPU (running at 1.80GHz). We performed 8560 runs of the GMDH algorithm with BTE, NAF and 950-NIBNAF. The last expansion is with the maximal possible w such that the resulting output

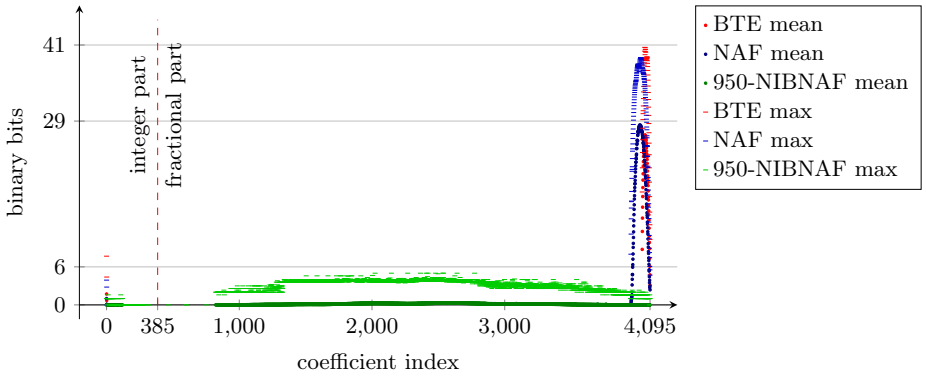

Fig. 7. The mean and the maximal size per coefficient of the resulting polynomial.

polynomial still has discernible integer and fractional parts. Correct evaluation of the prediction algorithm requires the plaintext modulus to be bigger than the maximal coefficient of the resulting polynomial. This lower bound for t can be deduced either from the maximal coefficient (in absolute value) appearing after any run or, in case of known distribution of coefficient values, from the mean and the standard deviation. In both cases increasing window sizes reduce the bound as depicted in Figure 7. Since negative encoding coefficients are used, 950-NIBNAF demands a plaintext modulus of 7 bits which is almost 6 times smaller than for BTE and NAF.

As expected, w-NIBNAF encodings have longer expansions for bigger w's and that disrupts the decoding procedure in [6,16]. Namely, they naively split the resulting polynomial into two parts of equal size. As one can observe in Figure 7, using 950-NIBNAF, decoding in this manner will not give correct results. Instead, the splitting index i_s should be shifted towards zero, i.e. to 385. To be specific [6, Lem. 1] states that i_s lies in the interval $(d_i + 1, d - d_f)$ where $d_i = 2^{r+1}(\ell(w)_i^{\text{inp}} + \ell(w)_i^{\text{pol}}) - \ell(w)_i^{\text{pol}}$ and $d_f = 2^{r+1}(\ell(w)_f^{\text{inp}} + \ell(w)_f^{\text{pol}}) - \ell(w)_f^{\text{pol}}$. Indeed, this is the worst case estimation which results in the maximal $w = 74$ for the current network configuration.

However the impact of the lower coefficients of the fractional part can be much smaller than the precision required by an application. In our use case the prediction value should be precise up to $e_f^{\text{inp}} = 1$. We denote the aggregated sum of lower coefficients multiplied by corresponding powers of the w-NIBNAF base as $L(j) = \sum_{i=j-1}^{i_s} a_i b_w^{-i}$. Then the omitted fractional coefficients a_i should satisfy $|L(i_c)| < 1$, where i_c is the index after which coefficients are ignored.

To find i_c we computed $L(j)$ for every index j of the fractional part and stored those sums for each run of the algorithm. For fixed j the distribution of $L(j)$ is bimodal with mean $\mu_{L(j)}$ and standard deviation $\sigma_{L(j)}$ (see Figure 8). Despite the fact that this unknown distribution is not normal, we naively approximate the prediction interval $[\mu_{L(j)} - 6\sigma_{L(j)}, \mu_{L(j)} + 6\sigma_{L(j)}]$ that will contain the future observation with high probability. It seems to be a plausible guess in

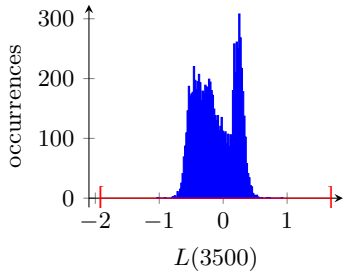

Fig. 8. The distribution of $L(3500)$ over 8560 runs of the GMDH algorithm and an approximation of its prediction interval in red.

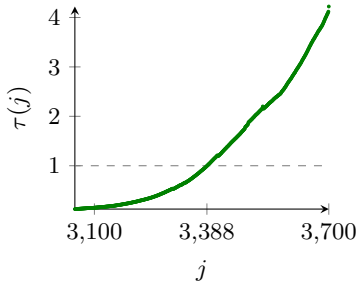

Fig. 9. The expected precision loss after ignoring fractional coefficients less than j.

	t	CRT factors	timing for one run
950-NIBNAF	$2^{5.044}$	1	2.57 s
BTE (this paper)	$2^{41.627}$	5	12.95 s
BTE [6]	$2^{103.787}$	13	32.5 s

Table 2. GMDH implementation with 950-NIBNAF and BTE [6]

this application because all observed $L(j)$ fall into that region with a big overestimate according to Figure 8. Therefore i_c is equal to the maximal j that satisfies $\tau(j) < 1$, where $\tau(j) = \max(|\mu_{L(j)} - 6\sigma_{L(j)}|, |\mu_{L(j)} + 6\sigma_{L(j)}|)$.

As Figure 9 shows, i_c is equal to 3388. Thus, the precision setting allows an overflow in any fractional coefficient a_j for $j < 3388$. The final goal is to provide the bound on t which is bigger than any a_j for $j \geq 3388$. Since the explicit distributions of coefficients are unknown and seem to vary among different indices, we rely in our analysis on the maximal coefficients occurring among all runs. Hence, the plaintext modulus should be bigger than $\max_{j \geq 3388}\{a_j\}$ over all resulting polynomials. Looking back at Figure 7, one can find that $t = 33$ suffices.

As mentioned above t is constrained in two ways: from the circuit and from the SHE correctness requirements. In our setup the ciphertext modulus is $q \approx 2^{186}$ and the standard deviation of noise is $\sigma = 102$, which together impose that $t \leq 396$ [6]. This is perfectly compatible with $t = 33$, therefore 950-NIBNAF allows us to omit the CRT trick and work with a single modulus, reducing the sequential timings by a factor 13. In the parallel mode it means that 13 times less memory is needed.

Additionally, these plaintext moduli are much smaller than the worst case estimation from Section 3. For 950-NIBNAF we take $d \in [542, 821]$ according to the encoding degrees of input data and network coefficients. Any such encoding contains only one non-zero coefficient. Consequently, any product of those encodings has only one non-zero coefficient which is equal to ± 1. When all mono-

mials of the GMDH polynomial result in an encoding with the same index of a non-zero coefficient, the maximal possible coefficient of the output encoding will occur. In this case the maximal coefficient is equal to the evaluation of the GMDH network with all input data and network coefficients being just 1. It leads to $t = 2 \cdot 6^{15} \simeq 2^{39.775}$.

One further consequence of smaller t is that one can reconsider the parameters of the underlying SHE scheme. Namely, one can take smaller q and σ that preserve the same security level and require a smaller bound on t instead of 396 taken above. Given $t = 33$ from above experiments, q reduces to 2^{154} together with $\sigma \approx 5$ that corresponds to smaller sizes of ciphertexts and faster SHE routines, where σ is taken the minimal possible to prevent the Arora-Ge attack [4] as long as each batch of input parameters is encrypted with a different key. Unfortunately, it is not possible to reduce the size of q by 32 bits in our implementation due to constraints of the FV-NFLlib library.

5 Conclusions

We have presented a generic technique to encode real numbers using a non-integral base. This encoding technique is especially suitable for use when evaluating homomorphic functions since it utilizes the large degree of the defining polynomial imposed by the security requirements. This leads to a considerably smaller growth of the coefficients and allows one to reduce the size of the plaintext modulus significantly, resulting in faster implementations. We show that in the setting studied in [6], where somewhat homomorphic function evaluation is used to achieve a privacy-preserving electricity forecast algorithm, the plaintext modulus can be reduced from about 2^{103} when using a balanced ternary expansion encoding, to $33 \simeq 2^{5.044}$ when using the encoding method introduced in this paper (non-integral base non-adjacent form with window size w), see Table 2. This smaller plaintext modulus means a factor 13 decrease in the running time of this privacy-preserving forecasting algorithm: closing the gap even further to making this approach suitable for industrial applications in the smart grid.

References

1. M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in helib and SEAL. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017*, volume 10211 of *LNCS*, pages 103–129, 2017.
2. I. Aliev. Siegel's lemma and sum-distinct sets. *Discrete Comput. Geom.*, 39(1-3):59–66, 2008.
3. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange – a new hope. In *USENIX Security Symposium*. USENIX Association, 2016.
4. S. Arora and R. Ge. New algorithms for learning in presence of errors. In L. Aceto, M. Henzinger, and J. Sgall, editors, *ICALP 2011, Part I*, volume 6755 of *LNCS*, pages 403–415. Springer, Heidelberg, July 2011.
5. C. Bootland. Central Extended Binomial Coefficients and Sums of Powers. In preparation.

6. J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren. Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling. In M. Joye and A. Nitaj, editors, *AFRICACRYPT 2017*, volume 10239 of *LNCS*, pages 184–201, 2017.
7. J. W. Bos, C. Costello, M. Naehrig, and D. Stebila. Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. In *IEEE S&P*, pages 553–570. IEEE Computer Society, 2015.
8. J. W. Bos, K. Lauter, J. Loftus, and M. Naehrig. Improved security for a ring-based fully homomorphic encryption scheme. In M. Stam, editor, *Cryptography and Coding 2013*, volume 8308 of *LNCS*, pages 45–64. Springer, 2013.
9. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, Jan. 2012.
10. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, Aug. 2011.
11. J. H. Cheon, J. Jeong, J. Lee, and K. Lee. Privacy-preserving computations of predictive medical models with minimax approximation and non-adjacent form. In *Proceedings of WAHC 2017*, LNCS, 2017.
12. J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. Cryptology ePrint Archive, Report 2016/421, 2016. `http://eprint.iacr.org/2016/421`.
13. H. Cohen, A. Miyaji, and T. Ono. Efficient Elliptic Curve Exponentiation Using Mixed Coordinates. In K. Ohta and D. Pei, editors, *Advances in Cryptology – ASIACRYPT '98*, volume 1514 of *LNCS*, pages 51–65. Springer, 1998.
14. Commission for Energy Regulation. Electricity smart metering customer behaviour trials (CBT) findings report. Technical Report CER11080a, 2011. `http://www.cer.ie/docs/000340/cer11080(a)(i).pdf`.
15. A. Costache, N. P. Smart, and S. Vivek. Faster homomorphic evaluation of Discrete Fourier Transforms. *IACR Cryptology ePrint Archive*, 2016.
16. A. Costache, N. P. Smart, S. Vivek, and A. Waller. Fixed point arithmetic in SHE schemes. In *SAC 2016*, LNCS. Springer, 2016.
17. CryptoExperts. FV-NFLlib. `https://github.com/CryptoExperts/FV-NFLlib`, 2016.
18. A. de Moivre. *The doctrine of Chances*. Woodfall, 1738.
19. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Manual for using homomorphic encryption for bioinformatics. Technical report, MSR-TR-2015-87, Microsoft Research, 2015.
20. N. Dowlin, R. Gilad-Bachrach, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In M. Balcan and K. Q. Weinberger, editors, *International Conference on Machine Learning*, volume 48, pages 201–210. JMLR.org, 2016.
21. S. Eger. Stirling's Approximation for Central Extended Binomial Coefficients. *The American Mathematical Monthly*, 121:344–349, 2014.
22. L. Euler. De evolutione potestatis polynomialis cuiuscunque $(1 + x + x^2 + x^3 + x^4 + \text{etc.})^n$. *Nova Acta Academiae Scientarum Imperialis Petropolitinae*, 12:47–57, 1801.
23. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

24. C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.
25. N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. A. Huss. On the design of hardware building blocks for modern lattice-based encryption schemes. In E. Prouff and P. Schaumont, editors, *CHES 2012*, volume 7428 of *LNCS*, pages 512–529. Springer, Heidelberg, Sept. 2012.
26. T. Güneysu, T. Oder, T. Pöppelmann, and P. Schwabe. Software speed records for lattice-based signatures. In P. Gaborit, editor, *PQCrypto 2013*, volume 7932 of *LNCS*, pages 67–82. Springer, 2013.
27. K. E. Lauter, A. López-Alt, and M. Naehrig. Private computation on encrypted genomic data. In D. F. Aranha and A. Menezes, editors, *LATINCRYPT 2014*, volume 8895 of *LNCS*, pages 3–27. Springer, Heidelberg, Sept. 2015.
28. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *MATH. ANN*, 261:515–534, 1982.
29. V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In K. Nyberg, editor, *FSE 2008*, volume 5086 of *LNCS*, pages 54–72. Springer, Heidelberg, Feb. 2008.
30. L. Mattner and B. Roos. Maximal probabilities of convolution powers of discrete uniform distributions. *Statistics & Probability Letters*, 78(17):2992 – 2996, 2008.
31. M. Naehrig, K. E. Lauter, and V. Vaikuntanathan. Can homomorphic encryption be practical? In C. Cachin and T. Ristenpart, editors, *ACM Cloud Computing Security Workshop – CCSW*, pages 113–124. ACM, 2011.
32. T. Pöppelmann and T. Güneysu. Towards practical lattice-based public-key encryption on reconfigurable hardware. In T. Lange, K. Lauter, and P. Lisonek, editors, *SAC 2013*, volume 8282 of *LNCS*, pages 68–85. Springer, Heidelberg, Aug. 2014.
33. G. W. Reitwiesner. *Binary Arithmetic*, volume 1 of *Advances in Computers*, pages 231–308. Academic Press, 1960.
34. D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 27–47. Springer, Heidelberg, May 2011.
35. J. W. Swanepoel. On a generalization of a theorem by Euler. *Journal of Number Theory*, 149:46–56, 2015.

A Proofs

Lemma 1 *For an integer $w \geq 1$ the polynomial $F_w(x) = x^{w+1} - x^w - x - 1$ has a unique positive root $b_w > 1$. The sequence $b_1, b_2, \ldots$ is strictly decreasing and $\lim_{w\to\infty} b_w = 1$. Further, $(x^2 + 1) \mid F_w(x)$ for $w \equiv 3 \mod 4$.*

Proof. For $w \geq 1$, $F'_w(x) = (w+1)x^w - wx^{w-1} - 1 = (x-1)((w+1)x^{w-1} + x^{w-2} + \cdots + 1)$ so that for $x \geq 0$ there is only one turning point of $F_w(x)$, at $x = 1$. Further, $F''_w(x) = (w+1)wx^{w-1} - w(w-1)x^{w-2}$, which takes the value $2w > 0$ at $x = 1$, so the turning point is a minimum. Since $F_w(0) = -1$ and $\lim_{x\to\infty} F_w(x) = \infty$ we conclude that there is a unique positive root of $F_w(x)$, $b_w > 1$, for any $w \geq 1$. Further, we have that $F_{w+1}(x) = xF_w(x) + x^2 - 1$ so that $F_{w+1}(b_w) = b_w^2 - 1 > 0$ so that $b_{w+1} < b_w$ and hence the sequence b_w is strictly decreasing and bounded below by 1 so must converge

to some limit, say $b_\infty \geq 1$. If $b_\infty > 1$ then as b_w is the positive solution to $x - 1 = (x+1)/x^w$ and, for $x \geq b_\infty > 1$, $\lim_{w\to\infty}(x+1)/x^w = 0$ we see that $b_\infty = \lim_{w\to\infty} b_w = 1$, a contradiction. Hence $b_\infty = 1$ as required. Finally we see that $F_w(x) = x(x-1)(x^{w-1}+1) - (x^2+1)$ and for $w = 4k+3$ that $x^{w-1}+1 = 1-(-x^2)^{2k+1} = (x^2+1)\sum_{i=0}^{2k}(-x^2)^i$ and hence $(x^2+1) \mid F_{4k+3}(x)$. □

Recall that to find a lower bound on the maximal absolute coefficient size we consider w-balanced ternary sequences and to each sequence (a_i) we have the corresponding polynomial $\sum_i a_i X^i$ in R_t. As we only look at the coefficients and their relative distances we can simply assume that to each w-balanced ternary sequence $c_0, c_1, \ldots, c_d$ of length $d+1$ we have the associated polynomial $c_0 + c_1 X + \ldots + c_d X^d$ of degree d. Multiplication of polynomials thus gives us a way of multiplying (finite) w-balanced ternary sequences. In the rest of this appendix we use the polynomial and sequence notation interchangeably.

Lemma 3 *The maximal absolute size of a term that can appear in the product of p arbitrary w-balanced ternary sequences of length $d+1$ is at least*

$$B_w(d,p) := \sum_{k=0}^{\lfloor \lfloor p\lfloor d/w\rfloor/2\rfloor/(\lfloor d/w\rfloor+1)\rfloor} (-1)^k \binom{p}{k}\binom{p-1+\lfloor p\lfloor d/w\rfloor/2\rfloor - k\lfloor d/w\rfloor - k}{p-1} .$$

Proof. Consider the product of p sequences all of which are equal to $m = 10\cdots010\cdots010\cdots0$ of length $d+1$, having $n := \lfloor d/w\rfloor + 1$ non-zero terms (all being 1) and between each pair of adjacent non-zero terms there are exactly $w-1$ zero terms. Note that n is the maximal number of non-zero terms possible. As polynomials we have that $m = \sum_{i=0}^{n-1} X^{iw} = \frac{1-X^{nw}}{1-X^w}$, and hence we have

$$\begin{aligned} m^p &= \left(\frac{1-X^{nw}}{1-X^w}\right)^p = (1-X^{nw})^p \cdot (1-X^w)^{-p} \\ &= \left(\sum_{i=0}^{p}(-1)^i\binom{p}{i}X^{inw}\right)\left(\sum_{j=0}^{\infty}\binom{p-1+j}{p-1}X^{jw}\right) \\ &= \sum_{\ell=0}^{\infty}\left(\sum_{k=0}^{\lfloor \ell/n\rfloor}(-1)^k\binom{p}{k}\binom{p-1+\ell-kn}{p-1}\right)X^{\ell w} , \end{aligned}$$

where we have used the substitution $(i,j) \to (k,\ell) = (i, in+j)$. Since we know that m^p has degree $p(n-1)w$ we can in fact change the infinite sum over ℓ to a finite one from $\ell = 0$ to $p(n-1)$. To give the tightest lower bound we look for the maximal coefficient of m^p. It is well known that this maximal coefficient occurs as the central coefficient, namely of x^ℓ where ℓ is any nearest integer to $p(n-1)/2$ and this gives us $B_w(d,p)$. □

Lemma 4 *Suppose w divides d, then $B_w(d,p)$ equals the maximal absolute size of a term that can be produced by taking the product of p arbitrary w-balanced ternary sequences of length $d+1$.*

Proof. Let $S_w(d,p)$ be the set of all sequences that are the product of p arbitrary w-balanced ternary sequences of length $d+1$. To prove the lemma we bound all the terms of any sequence in $S_w(d,p)$. For $i=0,\ldots,pd$ define

$$m_w(d,p,i) = \max\{\, |a_i| \;\mid\; a_i \text{ is the } i\text{'th term of a sequence in } S_w(d,p)\,\}\ .$$

Define $B_w(d,p,\ell) := \sum_{k=0}^{\lfloor \ell/n\rfloor}(-1)^k\binom{p}{k}\binom{p-1+\ell-kn}{p-1}$, the coefficient of $X^{\ell w}$ in m^p. We will prove by induction on p that $m_w(d,p,i)\le B_w(d,p,\lfloor i/w\rfloor)$. We will use the notation $C_i(f)$ for a polynomial f to denote the coefficient of X^i in $f(X)$; this is defined to be zero if $i>\deg(f)$ or $i<0$. Thus in this notation $B_w(d,p,\ell) = C_{\ell w}\left((1-X^{nw})^p/(1-X^w)^p\right)$. The base case $p=1$ is straight forward, all the $m_w(d,p,i)$ are equal to 1 by the definition of a w-balanced ternary sequence. We therefore suppose that $m_w(d,p-1,i)\le B_w(d,p-1,\lfloor i/w\rfloor)$ for $0\le i\le (p-1)d$. Consider a product of p w-balanced ternary sequences of length $d+1$. It can be written as $f(X)e(X)$ where $f(X)\in S_w(d,p-1)$ and $e(X)\in S_w(d,1)$. We know that if $f(X)=\sum_{i=0}^{(p-1)d}a_iX^i$ then $|a_i|\le m_w(d,p-1,i)$ and if $e(X)=\sum_{j=0}^{d}\alpha_jX^j$ that $(fe)(X)=f(X)e(X)=\sum_{k=0}^{pd}\left(\sum_{i=\max(0,k-d)}^{\min((p-1)d,k)}a_i\alpha_{k-i}\right)X^k$, and due to the form of $e(X)$ we see that $|C_k(fe)|\le\sum_{j=1}^{n_k}|a_{i_j}|\le\sum_{j=1}^{n_k}m_w(d,p-1,i_j)$ for some $n_k\le n$, $\max(0,k-d)\le i_1<i_2<\cdots<i_{n_k}\le\min((p-1)d,k)$ and $i_{j+1}-i_j\ge w$ for $j=1,\ldots,n_k-1$.

The final condition on the i_j implies that the $\lfloor i_j/w\rfloor$ are distinct and since $m_w(d,p-1,i)$ is bounded above by $B_w(d,p-1,\lfloor i/w\rfloor)$, which depends only on $\lfloor i/w\rfloor$, we can recast this as

$$|C_k(fe)|\le\sum_{j=1}^{n_k}B_w(d,p-1,\ell_j)=\sum_{j=1}^{n_k}C_{\ell_j w}\left(\left(\frac{1-X^{nw}}{1-X^w}\right)^{p-1}\right)$$

where $\max(0,\lfloor k/w\rfloor-(n-1))\le\ell_1<\ell_2<\cdots<\ell_{n_k}\le\min((p-1)(n-1),\lfloor k/w\rfloor)$ where we have used that $d/w=n-1$ is an integer.

Since $\lfloor k/w\rfloor-(\lfloor k/w\rfloor-(n-1))+1=n$ we see that to make n_k as large as possible the ℓ_j must be the (at most n) consecutive integers in this range subject also to $0\le\ell_1$ and $\ell_{n_k}\le(p-1)(n-1)$. Thus taking a maximum over all possible f and e we have

$$\begin{aligned}
m_w(d,p,k) &\le \sum_{\ell=\lfloor k/w\rfloor-(n-1)}^{\lfloor k/w\rfloor} C_{\ell w}\left(\left(\frac{1-X^{nw}}{1-X^w}\right)^{p-1}\right)\\
&= \sum_{j=0}^{n-1} C_{\lfloor k/w\rfloor w}\left(\left(\frac{1-X^{nw}}{1-X^w}\right)^{p-1}X^{w(n-1-j)}\right)\\
&= C_{\lfloor k/w\rfloor w}\left(\left(\frac{1-X^{nw}}{1-X^w}\right)^{p}\right) = B_w(d,p,\lfloor k/w\rfloor)\ ,
\end{aligned}$$

which proves the inductive step. To finish the proof we note as before that the maximal value of $B_w(d,p,\lfloor k/w\rfloor)$ for $0\le k\le pd$ is reached, for example, when $\lfloor k/w\rfloor=\lfloor p\lfloor d/w\rfloor/2\rfloor$ and in this case we have $B_w(d,p)$ as required. □