KU LEUVEN

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], Frederik Vercauteren[1,4]

[1] imec-COSIC, ESAT, KU Leuven, Leuven, Belgium
[2] NXP Semiconductors, Leuven, Belgium
[3] Laboratoire Paul Painlevé, Université de Lille-1, France
[4] Open Security Research, Shenzhen, China

September 28, 2017, Taipei, Taiwan

COSIC

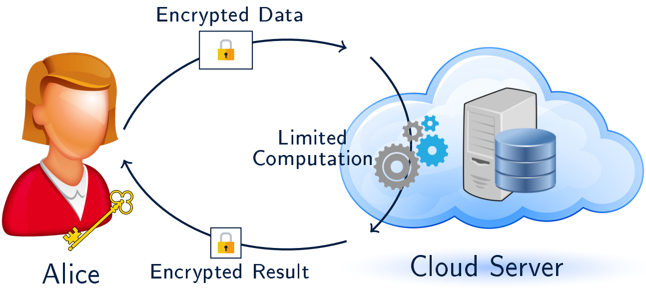
Somewhat Homomorphic Encryption (SHE)

KU LEUVEN

SHE: Cryptographic technique which allows an untrusted entity to perform a limited number of computational steps on encrypted data

KU LEUVEN

SHE: Cryptographic technique which allows an untrusted entity to perform a limited number of computational steps on encrypted data



The Plaintext Space – Mathematical Formulation

$$R_t := \frac{\mathbb{Z}_t[X]}{(X^{2^k}+1)}$$

- *$t \geq 2$ is a 'small' integer called the coefficient modulus*
- *$X^{2^k}+1$ is called the polynomial modulus*

This means polynomials of the form

$$a_0 + a_1 X + a_2 X^2 + \cdots + a_{2^k-1} X^{2^k-1}$$

where $0 \leq a_i < t$.

KU LEUVEN

The Plaintext Space – Mathematical Formulation

$$R_t := \frac{\mathbb{Z}_t[X]}{(X^{2^k}+1)}$$

- *$t \geq 2$ is a 'small' integer called the coefficient modulus*
- *$X^{2^k}+1$ is called the polynomial modulus*

This means polynomials of the form

$$a_0 + a_1 X + a_2 X^2 + \cdots + a_{2^k-1} X^{2^k-1}$$

where $0 \leq a_i < t$.

Problem

What is the best way to encode your data into the plaintext space?

Base b encoding - part 1

For integer b, $|b| > 1$, and real θ we can approximate θ by

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

for some integer coefficients a_i and integers r and s.

Base b encoding - part 1

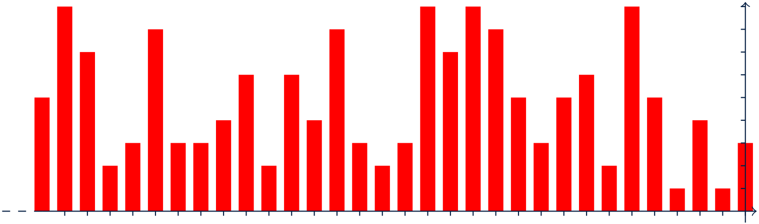
For integer b, $|b| > 1$, and real θ we can approximate θ by

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

for some integer coefficients a_i and integers r and s.

- Decimal expansion: here $b = 10$, $0 \le a_i \le 9$, together with a sign

$$\pi \approx 3.14159265358979323846264338327950 \cdots$$



Base b encoding - part 1

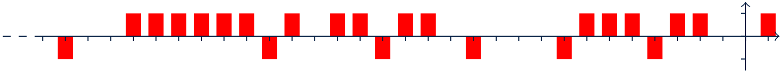
For integer b, $|b| > 1$, and real θ we can approximate θ by

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

for some integer coefficients a_i and integers r and s.

- Balanced ternary expansion: $b = 3$, $a_i \in \{-1, 0, 1\}$

$$\begin{aligned}\pi \approx\ & 3^1 + 3^{-2} + 3^{-3} - 3^{-4} + 3^{-5} + 3^{-6} + 3^{-7} - 3^{-8} - 3^{-12} \\ & + 3^{-14} + 3^{-15} - 3^{-16} + 3^{-17} + 3^{-18} + 3^{-20} - 3^{-21} + \cdots\end{aligned}$$



Base b encoding - part 1

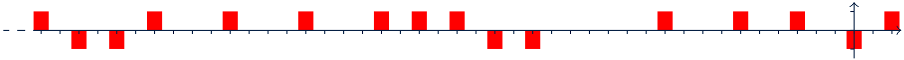
For integer b, $|b| > 1$, and real θ we can approximate θ by

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

for some integer coefficients a_i and integers r and s.

- Non-Adjacent Form (NAF) expansion: $b = 2$, $a_i \in \{-1, 0, 1\}$ and no two consecutive coefficients can both be non-zero

$$\pi \approx 2^2 - 2^0 + 2^{-3} + 2^{-6} + 2^{-10} - 2^{-17} - 2^{-19} + 2^{-21} + 2^{-23} + 2^{-25} + 2^{-29} + 2^{-33} + 2^{-37} - 2^{-39} - 2^{-41} + 2^{-43} + \cdots$$



Base b encoding - part 1

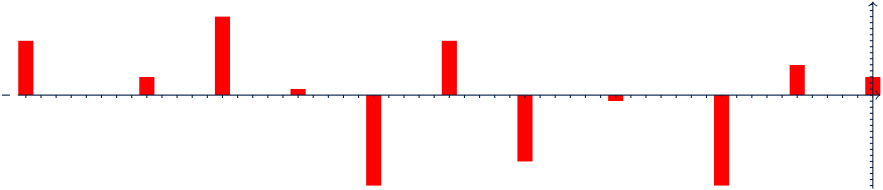
For integer b, $|b| > 1$, and real θ we can approximate θ by

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

for some integer coefficients a_i and integers r and s.

- w-NAF expansion for $w \geq 1$: $b = 2$, $|a_i| \leq 2^{w-1}$ either 0 or odd and no w consecutive coefficients can have more than one that non-zero

$$\pi \approx 3 \cdot 2^0 + 5 \cdot 2^{-5} - 15 \cdot 2^{-10} - 2^{-17} - 11 \cdot 2^{-23} + 9 \cdot 2^{-28}$$
$$- 15 \cdot 2^{-33} + 2^{-38} + 13 \cdot 2^{-43} + 3 \cdot 2^{-48} + 9 \cdot 2^{-56} + \cdots$$



Base b encoding - part 2

From the expression

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

one determines an encoding of θ by:

- replacing b by the indeterminate X
- reducing the a_i modulo t into $\left(-\frac{t}{2}, \frac{t}{2}\right]$
- reducing modulo $X^{2^k} + 1$

Base b encoding - part 2

From the expression

$$\theta \approx a_r b^r + a_{r-1} b^{r-1} + \cdots + a_1 b + a_0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \cdots + a_{-s} b^{-s}$$

one determines an encoding of θ by:

- replacing b by the indeterminate X
- reducing the a_i modulo t into $\left(-\frac{t}{2}, \frac{t}{2}\right]$
- reducing modulo $X^{2^k} + 1$

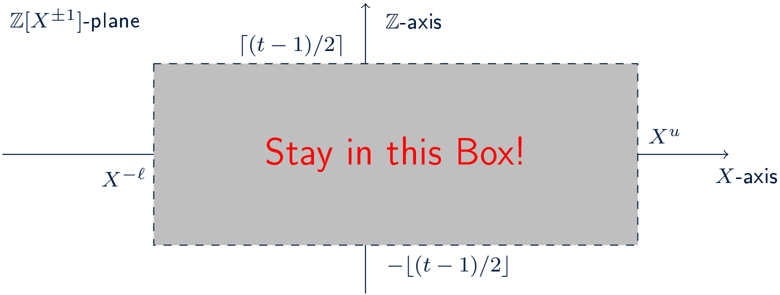
For encoding to be invertible we require:

- the range of possible a_i to be at most t
- there can be at most 2^k coefficients:
    - $r \leq u$ and $s \leq \ell$ where $\ell + u + 1 = 2^k$

Correct Decoding after Computation

Decoding is the inverse of encoding.

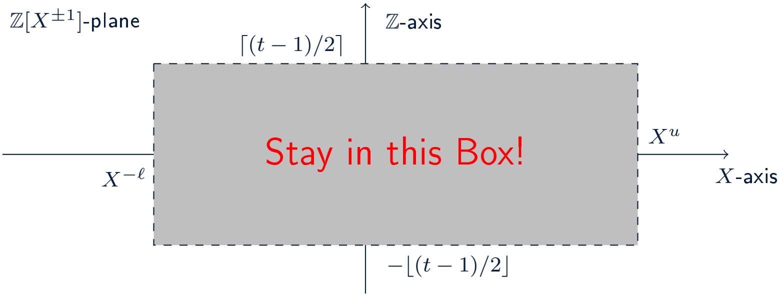
For correct decoding:



Correct Decoding after Computation

Decoding is the inverse of encoding.

For correct decoding:



If we know in advance what sorts of computations will be needed this places conditions on t, ℓ and u and thus 2^k.

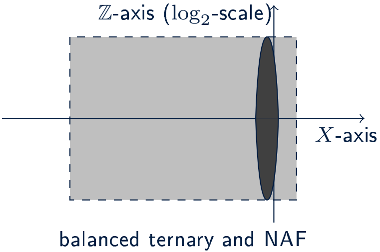
Constraints on t and 2^k

The precise constraints on t and 2^k depend on:

- The complexity of the computations
- The size and precision of the data and the type of expansion used
- Security and correctness requirements of the underlying SHE scheme
- Practicality of the scheme

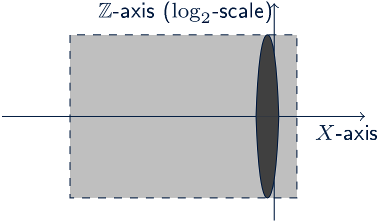
The Problem

Balanced ternary and NAF expansions do not make use of the whole plaintext space:

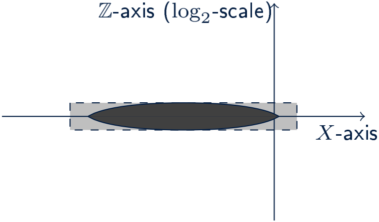


The Problem

Balanced ternary and NAF expansions do not make use of the whole plaintext space:



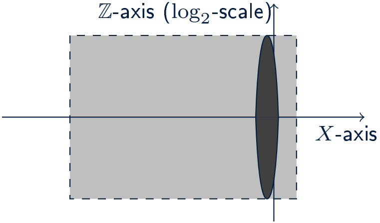
balanced ternary and NAF



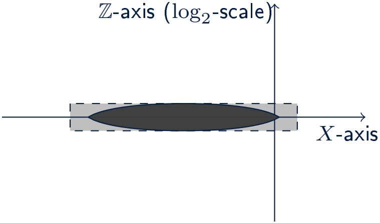
more optimal use of plaintext space

The Problem

Balanced ternary and NAF expansions do not make use of the whole plaintext space:



balanced ternary and NAF



more optimal use of plaintext space

Can we find a new encoding scheme that uses the plaintext space more efficiently?

Creating longer expansions

KU LEUVEN

A smaller base b gives in general:

- longer expansions
- smaller coefficients

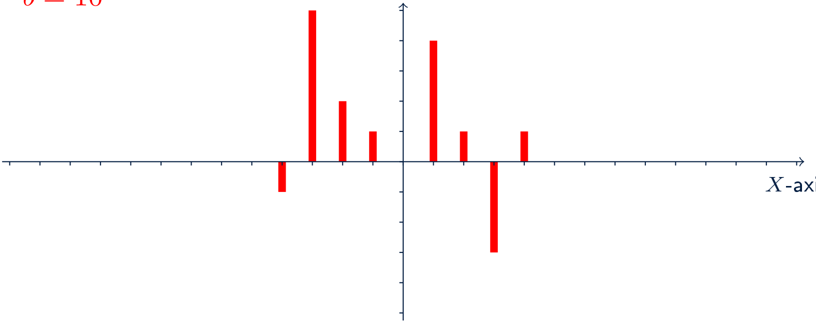
Creating longer expansions

KU LEUVEN

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 10$

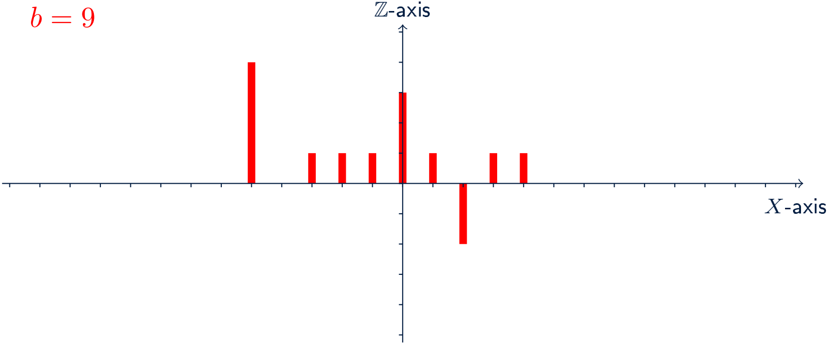


Encoding of 7140.1249 using a simple greedy algorithm using base b

Creating longer expansions

A smaller base b gives in general:

- longer expansions
- smaller coefficients

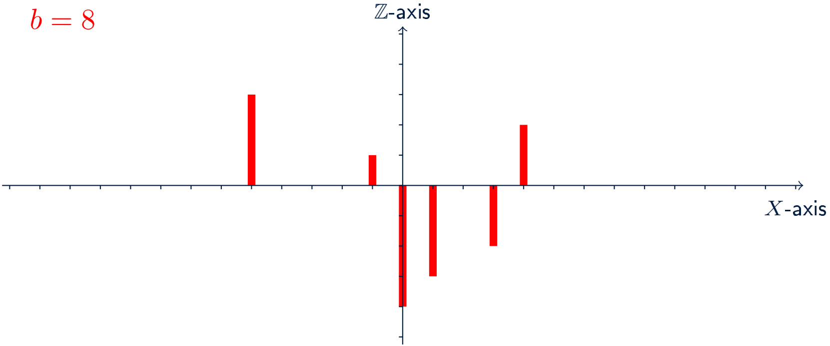


Encoding of 7140.1249 using a simple greedy algorithm using base b

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 8$



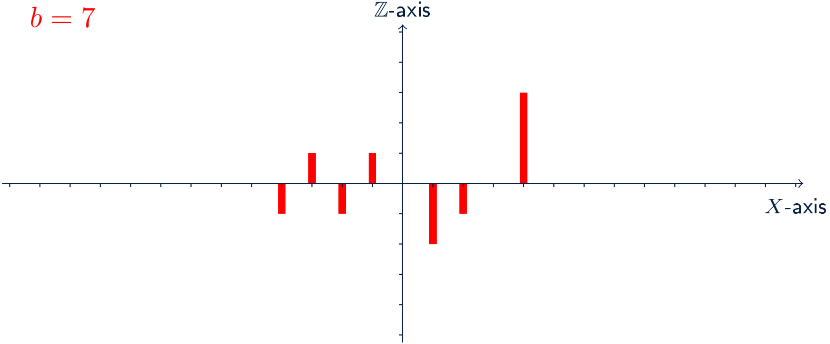
Encoding of 7140.1249 using a simple greedy algorithm using base b

Creating longer expansions

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 7$



Encoding of 7140.1249 using a simple greedy algorithm using base b

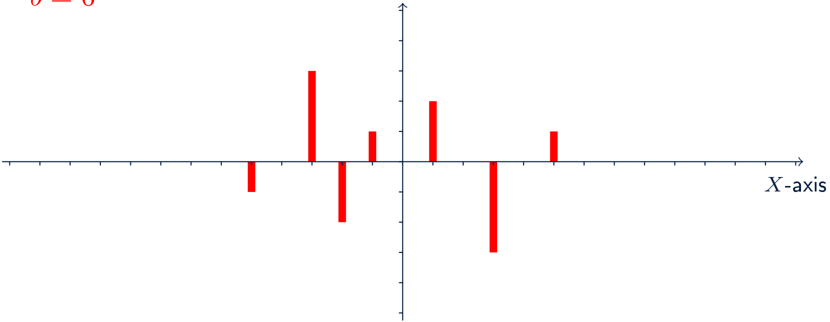
Creating longer expansions

KU LEUVEN

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 6$



Encoding of 7140.1249 using a simple greedy algorithm using base b

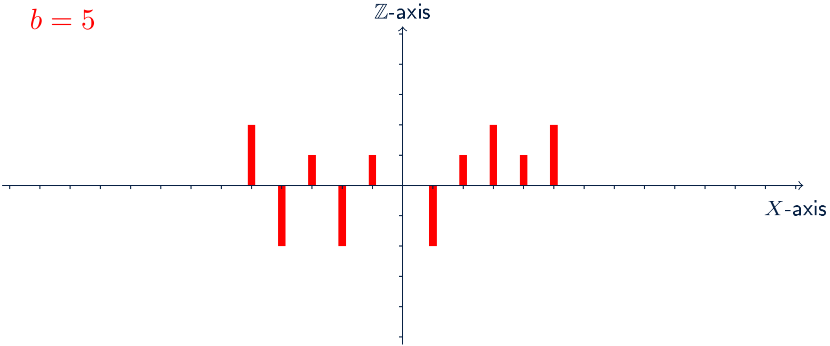
Creating longer expansions

KU LEUVEN

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 5$



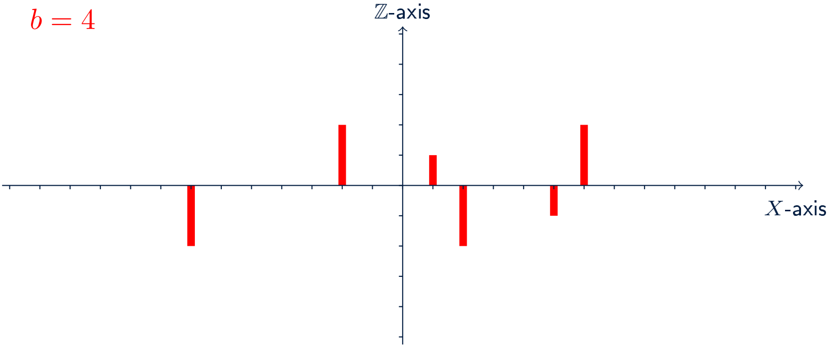
Encoding of 7140.1249 using a simple greedy algorithm using base b

Creating longer expansions

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 4$

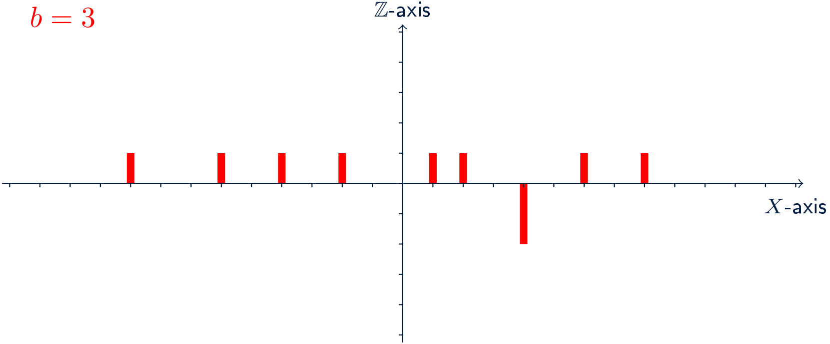


Encoding of 7140.1249 using a simple greedy algorithm using base b

Creating longer expansions

A smaller base b gives in general:

- longer expansions
- smaller coefficients



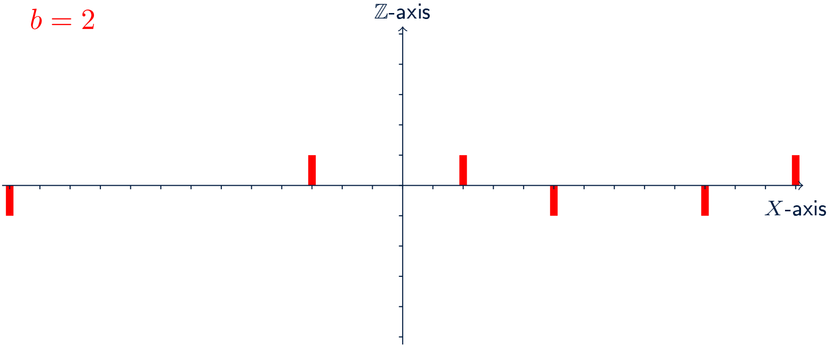
Encoding of 7140.1249 using a simple greedy algorithm using base b

Creating longer expansions

A smaller base b gives in general:

- longer expansions
- smaller coefficients

$b = 2$



Encoding of 7140.1249 using a simple greedy algorithm using base b

Idea: Use a non-integral base!

Also we would like:

- the coefficients to be as small as possible
- the expansions to be sparse
- a simple and efficient encoding algorithm

Making the base even smaller

Idea: Use a non-integral base!

Also we would like:

- the coefficients to be as small as possible
- the expansions to be sparse
- a simple and efficient encoding algorithm

This lead us to the idea of using a non-integral base non-adjacent form with window size w or ...

Making the base even smaller

Idea: Use a non-integral base!

Also we would like:

- the coefficients to be as small as possible
- the expansions to be sparse
- a simple and efficient encoding algorithm

This lead us to the idea of using a non-integral base non-adjacent form with window size w or ...

w-NIBNAF

w-NIBNAF

For a window size parameter $w \geq 1$ we define the base b_w as the unique positive real root of the polynomial

$$F_w(x) = x^{w+1} - x^w - x - 1$$

For a window size parameter $w \geq 1$ we define the base b_w as the unique positive real root of the polynomial

$$F_w(x) = x^{w+1} - x^w - x - 1$$

To encode a real value θ with base b_w to within precision ϵ:

1. If $|\theta| \leq \epsilon$ return 0
2. Find the closest signed power of b_w to θ, say σb_w^r where r is an integer and $\sigma \in \{\pm 1\}$
3. Recursively find the encoding of $\theta - \sigma b_w^r$, say this is $a(X)$
4. Return $\sigma X^r + a(X)$ as an encoding of θ

Analysing the Coefficient Growth

KU LEUVEN

- The size of the coefficient modulus t depends on how much the coefficients grow during computation
- Thus a good analysis of how the coefficients grow is required

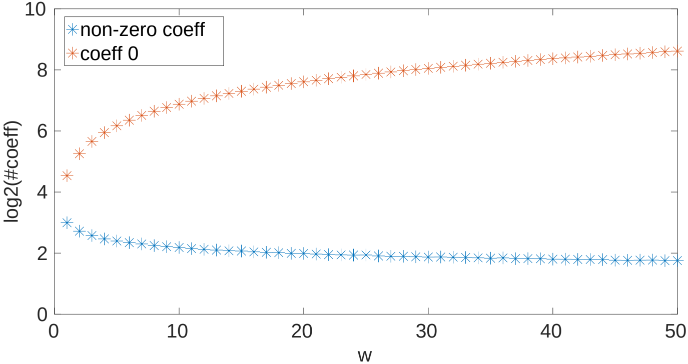
- The size of the coefficient modulus t depends on how much the coefficients grow during computation
- Thus a good analysis of how the coefficients grow is required

What affects the coefficient growth?

- The size of the coefficients in the original encodings
- The length of the original encodings
- The sparsity of the original encodings

Analysing the Coefficient Growth

For larger w's the expansions are longer but also sparser:



Plot of $\log_2$(#coefficients) against w averaged over $10\,000$ w-NIBNAF encodings of random integers in $[-2^{40}, 2^{40}]$

Overall the number of non-zero coefficients decreases as w increases

Multiplication - A Theoretical Worst Case Bound

$B_w(n,p) :=$ max coefficient that can appear after multiplying p encodings

where the encodings have at most n non-zero coefficients

$$n \approx (\text{max encoding length})/w + 1$$

Multiplication - A Theoretical Worst Case Bound

KU LEUVEN

$B_w(n,p) :=$ max coefficient that can appear after multiplying p encodings

where the encodings have at most n non-zero coefficients

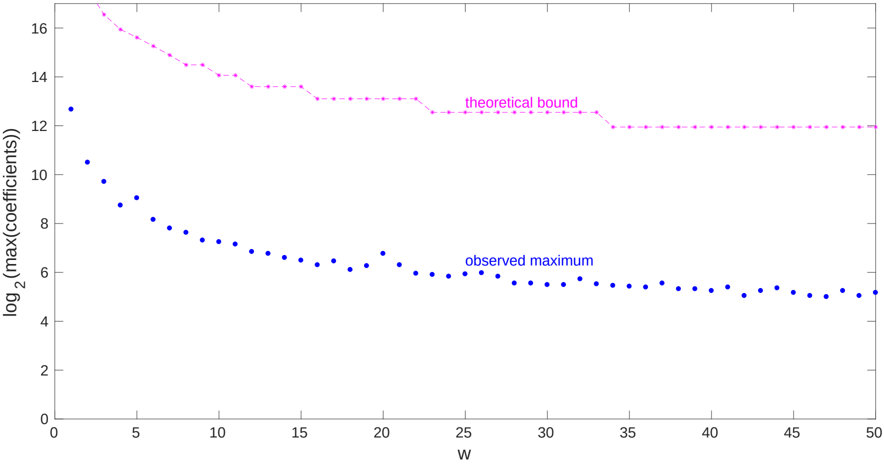
$$n \approx (\text{max encoding length})/w + 1$$

$$B_w(n,p) = \sum_{k=0}^{\lfloor \lfloor p(n-1)/2 \rfloor / n \rfloor} (-1)^k \binom{p}{k} \binom{p-1+\lfloor p(n-1)/2 \rfloor - kn}{p-1}$$

$$B_w(n,p) \leq \sqrt{\frac{6}{\pi p(n^2-1)}} n^p \quad \text{for } n \geq 2 \text{ and } p > 2^1$$

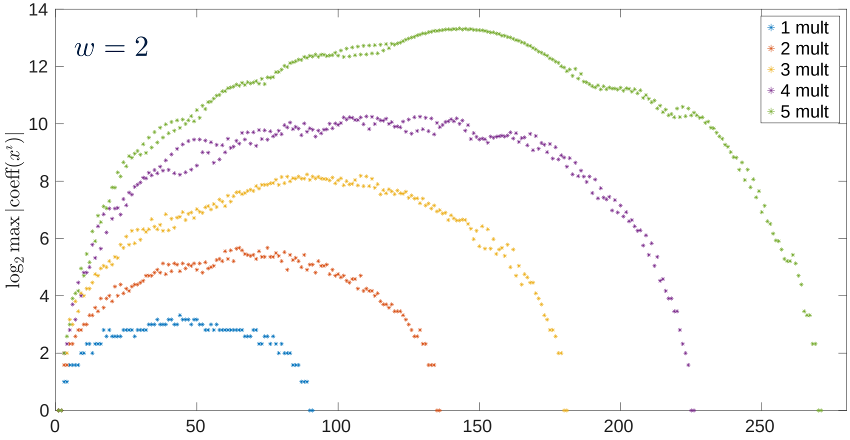
[1] L. Mattner and B. Roos. Maximal probabilities of convolution powers of discrete uniform distributions

In practice the theoretical worst case analysis is very pessimistic:



maximum absolute value of a coefficient after multiplying 5 w-NIBNAF encodings of random numbers in $\left[-2^{40}, 2^{40}\right]$ against w

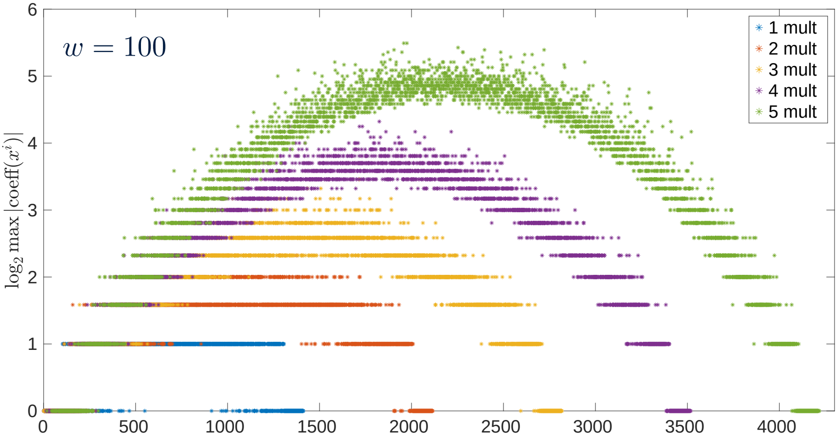
Multiplication - Filling the plaintext space



$\log_2$ of the maximum absolute value of the coefficient of x^i seen in 10 000 products of $2, 3, 4, 5$ and 6 2-NIBNAF encodings of random numbers in $[-2^{40}, 2^{40}]$ against i

Multiplication - Filling the plaintext space

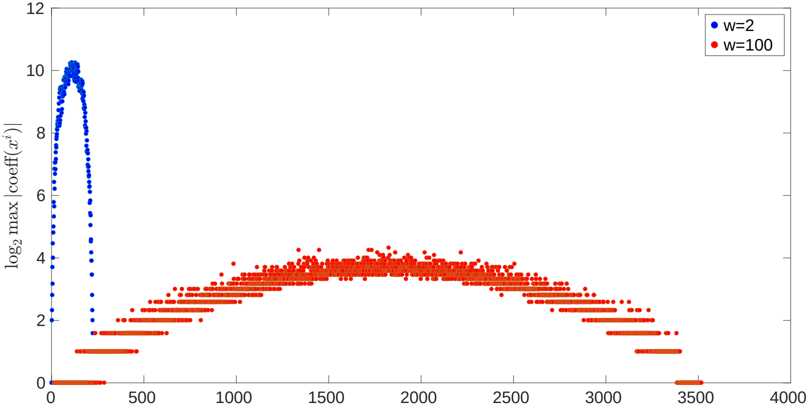
KU LEUVEN



$\log_2$ of the maximum absolute value of the coefficient of x^i seen in 10 000 products of $2, 3, 4, 5$ and 6 100-NIBNAF encodings of random numbers in $\left[-2^{40}, 2^{40}\right]$ against i

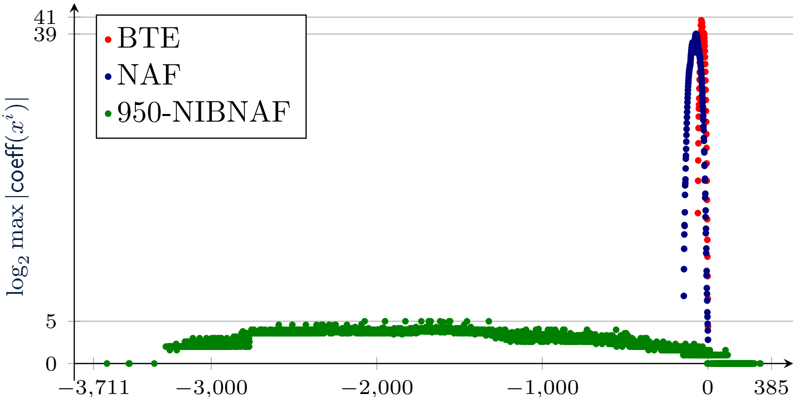
Multiplication - Filling the plaintext space

KU LEUVEN



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

Larger w give wider but flatter curves



$\log_2$ of the observed maximum absolute value of the coefficient of x^i during a privacy preserving forecasting algorithm for the electricity market

With $t = 33$ we are 13x faster!

Take home messages

- A new encoding technique called w-NIBNAF
- Encodes real numbers for use with HE schemes
- Uses smaller non-integral bases
- Gives longer expansions
- Much better use of the plaintext space
- Use a smaller value of the coefficient modulus t
- Smaller ciphertexts and faster implementations

Take home messages

KU LEUVEN

- A new encoding technique called w-NIBNAF
- Encodes real numbers for use with HE schemes
- Uses smaller non-integral bases
- Gives longer expansions
- Much better use of the plaintext space
- Use a smaller value of the coefficient modulus t
- Smaller ciphertexts and faster implementations

Thank you for listening!
Any questions?