# On Polynomial Functions Modulo $p^e$ and Faster Bootstrapping for Homomorphic Encryption

**Robin Geelen**[1], Ilia Iliashenko[2], Jiayi Kang[1], and Frederik Vercauteren[1]

[1]imec-COSIC, KU Leuven, and [2]CipherMode Labs

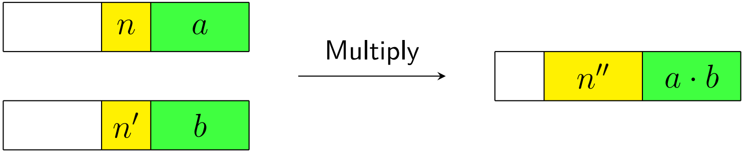
Eurocrypt 2023, April 24

# Fully Homomorphic Encryption

- Addition and multiplication over ciphertext space
  - $\mathsf{Enc}(a+b) = \mathsf{Enc}(a) + \mathsf{Enc}(b)$
  - $\mathsf{Enc}(a \cdot b) = \mathsf{Enc}(a) \cdot \mathsf{Enc}(b)$
- BGV and BFV scheme: $a + b$ and $a \cdot b$ computed over $\mathbb{Z}_{p^e}$
  - Prime number $p$ and positive integer $e$
- Complicated functions evaluated as polynomials

# Ciphertext Noise

- FHE ciphertexts are noisy
- Noise grows with homomorphic operations

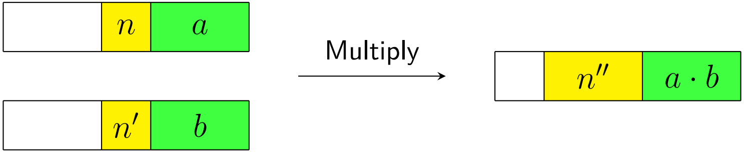





KU LEUVEN

# Ciphertext Noise

- FHE ciphertexts are noisy
- Noise grows with homomorphic operations




- Bootstrapping reduces noise

# Bootstrapping

Two core components of BGV and BFV bootstrapping:

- Linear transformations
- **Digit removal procedure**
  - Bottleneck in terms of speed and noise: $3\times$ to $50\times$ more expensive
  - Repeated polynomial evaluation

# Some Terminology

## Polyfunctions

Function $f : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e}$ is a *polyfunction* if there exists $F(X) \in \mathbb{Z}[X]$ s.t.

$$F(a) = f(a) \pmod{p^e}$$

for each $a \in \mathbb{Z}$. We call $F(X)$ a *representation* of $f$.

# Some Terminology

## Polyfunctions

Function $f : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e}$ is a *polyfunction* if there exists $F(X) \in \mathbb{Z}[X]$ s.t.

$$F(a) = f(a) \pmod{p^e}$$

for each $a \in \mathbb{Z}$. We call $F(X)$ a *representation* of $f$.

If $e = 1$

- $\mathbb{Z}_{p^e}$ is a field
- Every function is a polyfunction
- Unique lowest-degree representation
  - Interpolation gives $F(X)$

# Some Terminology

## Polyfunctions

Function $f : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e}$ is a *polyfunction* if there exists $F(X) \in \mathbb{Z}[X]$ s.t.

$$F(a) = f(a) \pmod{p^e}$$

for each $a \in \mathbb{Z}$. We call $F(X)$ a *representation* of $f$.

If $e = 1$

- $\mathbb{Z}_{p^e}$ is a field
- Every function is a polyfunction
- Unique lowest-degree representation
  - Interpolation gives $F(X)$

If $e > 1$

- $\mathbb{Z}_{p^e}$ is not a field
- Not every function is a polyfunction
- No unique representation

# Objectives of This Work

- Systematic study of polyfunctions
  - How to determine whether a function is a polyfunction?
  - How to obtain a representation of a polyfunction?
  - How to find FHE-friendly representations?
    - Less noise growth
    - Fewer scalar and non-scalar multiplications
- Accelerate bootstrapping for BGV and BFV
  - Focus on digit removal procedure

# Digit Extraction Function

- Digit removal procedure is built from digit extraction function

## Digit Extraction Function

Denote by $w_0$ the least significant digit of $w \in \mathbb{Z}_{p^e}$ in its base-$p$ expansion, then *digit extraction* is the map

$$g_e : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e} : w \mapsto w_0$$

$$\underbrace{{\color{purple}\blacksquare} \cdots {\color{cyan}\blacksquare}{\color{red}\blacksquare}}_{e \text{ digits}} \mapsto \underbrace{0 \cdots 0{\color{red}\blacksquare}}_{e \text{ digits}}$$

# Digit Extraction Function

- Digit removal procedure is built from digit extraction function

## Digit Extraction Function

Denote by $w_0$ the least significant digit of $w \in \mathbb{Z}_{p^e}$ in its base-$p$ expansion, then *digit extraction* is the map

$$g_e : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e} : w \mapsto w_0$$

$$\underbrace{\blacksquare \cdots \blacksquare\blacksquare}_{e \text{ digits}} \mapsto \underbrace{0 \cdots 0\blacksquare}_{e \text{ digits}}$$

- Digit extraction $g_e$ is a polyfunction with representation $G_e(X)$

# Representations of the Digit Extraction Function

## Representations of $g_e$ for $p = 2$ and $e = 8$

- Halevi and Shoup perform repeated squaring and find

$$G_8^{HS}(X) = X^{2^7} \pmod{2^8}$$

- Chen and Han find a lowest degree representation

$$G_8^{CH}(X) = 13X^8 + 96X^7 + 84X^6 + 32X^5 + 32X^4 \pmod{2^8}$$

# Representations of the Digit Extraction Function

## Representations of $g_e$ for $p = 2$ and $e = 8$

- Halevi and Shoup perform repeated squaring and find

$$G_8^{HS}(X) = X^{2^7} \pmod{2^8}$$

- Chen and Han find a lowest degree representation

$$G_8^{CH}(X) = 13X^8 + 96X^7 + 84X^6 + 32X^5 + 32X^4 \pmod{2^8}$$

Their difference satisfies $\underbrace{G_8^{HS}(X) - G_8^{CH}(X)}_{\text{Null polynomial}} \equiv 0 \pmod{2^8}$

# Null Polynomials and Equivalent Representations

- Polynomial $O(X)$ that evaluates the zero function modulo $p^e$ is called a **null polynomial**:

$$g_e \iff \Big\{ G_e(X) + O(X) \Big\}$$



KU LEUVEN

# Null Polynomials and Equivalent Representations

- Polynomial $O(X)$ that evaluates the zero function modulo $p^e$ is called a **null polynomial**:

$$g_e \quad \Longleftrightarrow \quad \Big\{G_e(X) + O(X)\Big\}$$

Observation: obtain equivalent representations by adding null polynomials
$\Rightarrow$ Select **FHE-friendly representation**

But how to find these null polynomials?

# Finding Null Polynomials

- Trivial for $e = 1$: Fermat's little theorem states that $X^p - X$ is a null polynomial modulo $p$

# Finding Null Polynomials

- Trivial for $e = 1$: Fermat's little theorem states that $X^p - X$ is a null polynomial modulo $p$
- More complicated for $e > 1$:
  - Define **falling factorial polynomials**: $(X)_i = X(X-1) \cdot \ldots \cdot (X-i+1)$
  - Evaluation of $(X)_i$ at any integer is divisible by $i!$

# Finding Null Polynomials

- Trivial for $e = 1$: Fermat's little theorem states that $X^p - X$ is a null polynomial modulo $p$
- More complicated for $e > 1$:
  - Define **falling factorial polynomials**: $(X)_i = X(X-1) \cdot \ldots \cdot (X-i+1)$
  - Evaluation of $(X)_i$ at any integer is divisible by $i!$
  - The set of all null polynomials includes
    - $(X)_i$ if $i!$ is divisible by $p^e$
    - $p^{e-\nu_p(i!)} \cdot (X)_i$ otherwise
    - Linear combinations of the above

# Lowest Degree Representation

- Let $O(X)$ be a monic null polynomial of the lowest degree
- Apply **Euclidean division** on any representation $G_e(X)$:

$$G_e(X) = O(X) \cdot Q(X) + G'_e(X)$$

- Gives another representation $G'_e(X)$ of degree less than $\deg(O(X)) \leqslant p \cdot e$

# Lowest Degree Representation

- Let $O(X)$ be a monic null polynomial of the lowest degree
- Apply **Euclidean division** on any representation $G_e(X)$:

$$G_e(X) = O(X) \cdot Q(X) + G'_e(X)$$

- Gives another representation $G'_e(X)$ of degree less than $\deg(O(X)) \leqslant p \cdot e$

## Chen/Han representation of $g_e$

- Chen/Han representation $G_e^{CH}(X)$ has minimal degree $(p-1) \cdot (e-1) + 1$
- Still we can search for even better representations



KU LEUVEN

# Improvement I: Parity

- Digit extraction is a symmetric function
  - If $p = 2$: $g_e(-a) = g_e(a)$
  - If $p > 2$: $g_e(-a) = -g_e(a)$

# Improvement I: Parity

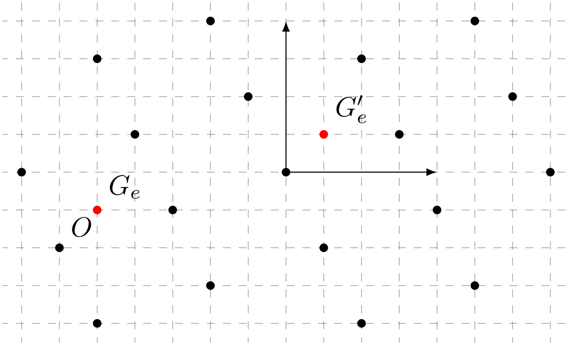
- Digit extraction is a symmetric function
  - If $p = 2$: $g_e(-a) = g_e(a)$
  - If $p > 2$: $g_e(-a) = -g_e(a)$
- Choose representation with only **even- or odd-exponent** terms
  - For $p > 2$: $G_e(X) = (G_e^{CH}(X) - G_e^{CH}(-X))/2$
  - The case $p = 2$ is more tricky: see paper

# Improvement I: Parity

- Digit extraction is a symmetric function
  - If $p = 2$: $g_e(-a) = g_e(a)$
  - If $p > 2$: $g_e(-a) = -g_e(a)$
- Choose representation with only **even- or odd-exponent** terms
  - For $p > 2$: $G_e(X) = (G_e^{CH}(X) - G_e^{CH}(-X))/2$
  - The case $p = 2$ is more tricky: see paper
- Compared to Chen/Han, we have the following complexity gain:
  - $\times 1/\sqrt{2}$ non-scalar multiplications
  - $\times 1/2$ scalar multiplications

# Improvement II: Lattice

- Interpreting polynomials as coefficient vectors, null polynomials with degree bound $n$ form an $n+1$-dimensional lattice
- Solve **closest vector problem**: $G_e'(X) = G_e(X) - O(X)$





KU LEUVEN

# Example

## Representations of $g_e$ for $p = 2$ and $e = 8$

- Recall that Chen and Han find a lowest degree representation

$$G_8^{CH}(X) = 13X^8 + 96X^7 + 84X^6 + 32X^5 + 32X^4 \pmod{2^8}$$

- Improvement I and II result in

$$G_8(X) = 13X^8 - 12X^6 \pmod{2^8}$$

# Improvement III: Function Composition

Idea: decompose digit extraction function as $g_e = g_{e,e'} \circ g_{e'}$ for some $e' < e$

$$g_e : \underbrace{\blacksquare \cdots \blacksquare\blacksquare \cdots \blacksquare\blacksquare}_{e \text{ digits}} \xmapsto{g_{e'}} * \cdots * \underbrace{0 \cdots 0\blacksquare}_{e' \text{ digits}}$$

# Improvement III: Function Composition

Idea: decompose digit extraction function as $g_e = g_{e,e'} \circ g_{e'}$ for some $e' < e$

$$g_e : \underbrace{\blacksquare \cdots \blacksquare\,\blacksquare \cdots \blacksquare\,\blacksquare}_{e \text{ digits}} \xmapsto{g_{e'}} * \cdots * \underbrace{0 \cdots 0\,\blacksquare}_{e' \text{ digits}} \xmapsto{g_{e,e'}} 0 \cdots 0\; 0 \cdots 0\,\blacksquare$$

- Relevant domain of $g_{e,e'}$ is Range$(g_{e'}) \subset \mathbb{Z}_{p^e}$
  $\Rightarrow$ More null polynomials defined over this range

# Improvement III: Function Composition

Idea: decompose digit extraction function as $g_e = g_{e,e'} \circ g_{e'}$ for some $e' < e$

$$g_e : \underbrace{\blacksquare \cdots \blacksquare \blacksquare \cdots \blacksquare \blacksquare}_{e \text{ digits}} \xmapsto{g_{e'}} * \cdots * \underbrace{0 \cdots 0 \blacksquare}_{e' \text{ digits}} \xmapsto{g_{e,e'}} 0 \cdots 0 \; 0 \cdots 0 \blacksquare$$

- Relevant domain of $g_{e,e'}$ is $\text{Range}(g_{e'}) \subset \mathbb{Z}_{p^e}$
  $\Rightarrow$ More null polynomials defined over this range
- Compared to Chen/Han, we have the following complexity gain:
  - Non-scalar multiplications: $\mathcal{O}(\sqrt{pe}) \Rightarrow \mathcal{O}(\sqrt{p}\sqrt[4]{e})$
  - Scalar multiplications: $\mathcal{O}(pe) \Rightarrow \mathcal{O}(p\sqrt{e})$
- Total degree increases with roughly a factor $p$

# Example

Function composition for $p = 2$, $e = 25$ and $e' = 8$

- Recall that improvement I and II result in

$$G_8(X) = 13X^8 - 12X^6 \pmod{2^8}$$

- Starting from $G_8(X)$, digit extraction modulo $2^{25}$ can be done with

$$G_{25,8}(X) = 6X^5 - 15X^4 + 10X^3 \pmod{2^{25}}$$

$\Rightarrow$ The composition $G_{25,8}(G_8(X))$ gives $g_{25}$

# The Digit Removal Procedure

Consider $w \in \mathbb{Z}_{p^e}$:

$$w = \underbrace{\textcolor{purple}{\blacksquare} \cdots \textcolor{yellow}{\blacksquare}\,\textcolor{pink}{\blacksquare} \cdots \textcolor{cyan}{\blacksquare}\,\textcolor{red}{\blacksquare}}_{e \text{ digits}}$$

Goal of digit removal:

$$\textcolor{purple}{\blacksquare} \cdots \textcolor{yellow}{\blacksquare}\,\underbrace{0 \cdots 0\ 0}_{v \text{ digits}}$$

# The Digit Removal Procedure

Consider $w \in \mathbb{Z}_{p^e}$:

$$w = \underbrace{{\color{purple}\blacksquare} \cdots {\color{yellow}\blacksquare}{\color{orange}\blacksquare} \cdots {\color{cyan}\blacksquare}{\color{red}\blacksquare}}_{e \text{ digits}}$$

Goal of digit removal:

$${\color{purple}\blacksquare} \cdots {\color{yellow}\blacksquare} \underbrace{0 \cdots 0\ 0}_{v \text{ digits}}$$

This requires

$$\begin{aligned}
w_0 &= 0 \cdots 0\ 0 \cdots 0\ {\color{red}\blacksquare} \\
w_1 &= 0 \cdots 0\ 0 \cdots {\color{cyan}\blacksquare} \\
&\vdots \\
w_{v-1} &= 0 \cdots 0\ {\color{orange}\blacksquare}
\end{aligned}$$

# Minimizing the Noise Growth

Besides from

$$w_0 = 0 \cdots \cdots 0\ 0\ 0 \blacksquare$$

One also needs to compute

$$\begin{aligned}
w_{0,1} &= * \cdots \cdots *\ *\ 0 \blacksquare \\
w_{0,2} &= * \cdots \cdots * 0\ 0 \blacksquare \\
&\vdots \\
w_{0,v-1} &= * \cdots * 0 \cdots\ 0 \blacksquare
\end{aligned}$$

# Three Versions of Digit Removal

## Halevi/Shoup

- Only use $G_e^{HS}(X)$
- Degree $p^{e-1}$

## Chen/Han

- Use $G_e^{HS}(X)$ and $G_e^{CH}(X)$
- Degree $(e - v) \cdot p^v$

## Our approach

- Only use our optimized representations $G_e(X)$
- Reuse polynomial evaluations while keeping the **same degree** as the Chen/Han version
- Evaluate multiple polynomials simultaneously in the same point using the baby-step/giant-step technique

# Experimental Results for Packed Bootstrapping

Original method / Our method

| | | | | |
|---|---|---|---|---|
| Cyclotomic index $m$ | | $127 \cdot 337$ | $101 \cdot 451$ | $43 \cdot 757$ |
| Params $(p, v, e)$ | | $(2, 7, 15)$ | $(17, 2, 6)$ | $(127, 2, 4)$ |
| Number of digit removals | | 21 | 40 | 14 |
| Remaining capacity (bits) | | 744/753 | 448/475 | 323/282 |
| Execution time (sec) | Linear maps | 134 | 150 | 290 |
| | Digit extract | 2014/743 | 2665/1879 | 1407/863 |
| | Total | 2248/877 | 2815/2029 | 1697/1153 |
| **Bootstrapping speedup** | | **2.6×** | **1.4×** | **1.5×** |

# Experimental Results for Digit Removal

Original method / No function composition / Function composition

| | | |
|---|---|---|
| Cyclotomic index $m$ | $42799$ | $63973$ |
| Params $(p, v, e, e')$ | $(2, 8, 59, 16)$ | $(3, 5, 37, 6)$ |
| Used capacity (bits) | $1049/991/1006$ | $1142/1047/1170$ |
| Execution time (sec) | $180/100/64$ | $191/151/119$ |
| **Digit removal speedup** | $\mathbf{1.8}\times/\mathbf{2.8}\times$ | $\mathbf{1.3}\times/\mathbf{1.6}\times$ |

# Conclusion

- Speed up bootstrapping for BGV and BFV up to $2.6\times$
- Better understanding of polyfunctions modulo $p^e$
  - Optimizations due to the existence of non-trivial null polynomials
  - Also of independent interest in cryptography

Thank you for your attention!