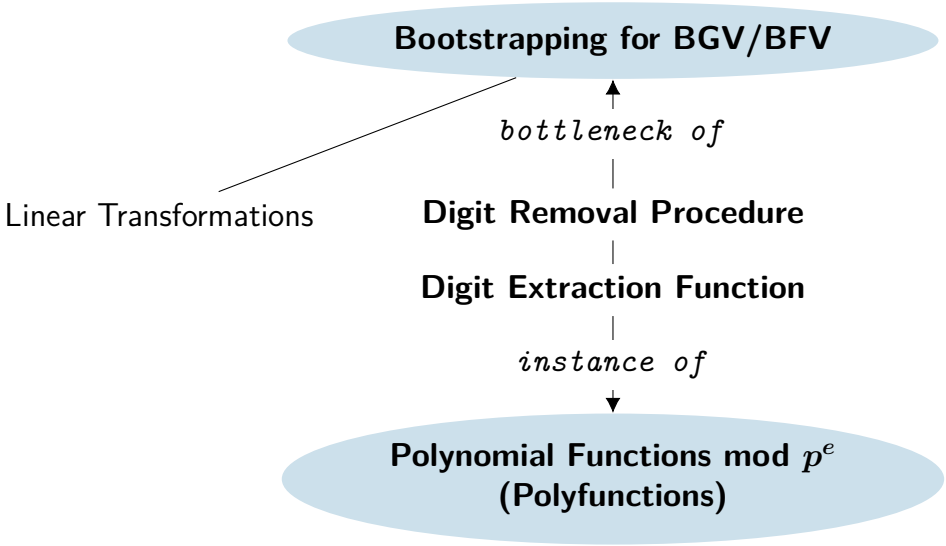
KU LEUVEN

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
FHE.org Conference, March 26, 2023

Bootstrapping and Polyfunctions



Polyfunctions

- Consider $f : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e}$, where p is a prime and e is a positive integer
- If there exists a polynomial $F(X) \in \mathbb{Z}[X]$ satisfying

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

then f is a **polyfunction** modulo p^e, and $F(X)$ is a **representation** of f

Polyfunctions

- Consider $f : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e}$, where p is a prime and e is a positive integer
- If there exists a polynomial $F(X) \in \mathbb{Z}[X]$ satisfying

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

then f is a **polyfunction** modulo p^e, and $F(X)$ is a **representation** of f

$e = 1$

- $\mathbb{Z}_{p^e}$ is a field
- Every function is a polyfunction
  - Interpolation $\Rightarrow F(X)$

Polyfunctions

- Consider $f : \mathbb{Z}_{p^e} \to \mathbb{Z}_{p^e}$, where p is a prime and e is a positive integer
- If there exists a polynomial $F(X) \in \mathbb{Z}[X]$ satisfying

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

then f is a **polyfunction** modulo p^e, and $F(X)$ is a **representation** of f

$e = 1$

- $\mathbb{Z}_{p^e}$ is a field
- Every function is a polyfunction
  - Interpolation $\Rightarrow F(X)$

$e > 1$

- $\mathbb{Z}_{p^e}$ is **not** a field
- Not every function is a polyfunction

Objectives of This Work

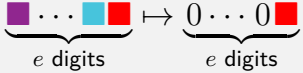
- Systematic study of polyfunctions
  - How to determine whether a function is a polyfunction?
  - How to obtain a representation of a polyfunction?
  - How to find **HE-friendly representations**?
    - Lower multiplicative depth
    - Fewer scalar and non-scalar multiplications
- Accelerate bootstrapping for BGV and BFV
  - Focus on digit extraction function and digit removal procedure

Digit Extraction Function

Digit Extraction Function

Denote the *symmetric digit decomposition*† of $w \in \mathbb{Z}_{p^e}$ by $w = \sum_{i=0}^{e-1} w_i p^i$, then digit extraction is the map

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



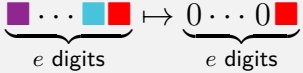
†Symmetric means $\begin{cases} w_i \in \{0, 1\} & \text{if } p = 2 \\ w_i \in \{-(p-1)/2, \ldots, (p-1)/2\} & \text{if } p > 2 \end{cases}$

Digit Extraction Function

Digit Extraction Function

Denote the *symmetric digit decomposition*† of $w \in \mathbb{Z}_{p^e}$ by $w = \sum_{i=0}^{e-1} w_i p^i$, then digit extraction is the map

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



†Symmetric means $\begin{cases} w_i \in \{0, 1\} & \text{if } p = 2 \\ w_i \in \{-(p-1)/2, \ldots, (p-1)/2\} & \text{if } p > 2 \end{cases}$

- Digit extraction g_e is a polyfunction, and G_e refers to its representation

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 the following 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 the following 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 Complete Representations

- Polynomial $O(X) \in \mathbb{Z}[X]$ that evaluates to the zero function modulo p^e is a **null polynomial**
- Let $\mathcal{O}_{p^e}$ be the set of all null polynomials modulo p^e

Polyfunction:		**Representations:**
g_e	$\Longleftrightarrow$	$\left\{G_e(X) + \mathcal{O}_{p^e}\right\}$

Null Polynomials and Complete Representations

- Polynomial $O(X) \in \mathbb{Z}[X]$ that evaluates to the zero function modulo p^e is a **null polynomial**
- Let $\mathcal{O}_{p^e}$ be the set of all null polynomials modulo p^e

Polyfunction:		**Representations:**
g_e	$\Longleftrightarrow$	$\left\{G_e(X) + \mathcal{O}_{p^e}\right\}$

Idea: select an **HE-friendly representation**

Null Polynomials

- Evaluating the **falling factorial polynomial**

$$(X)_i = X(X-1)\cdot\ldots\cdot(X-i+1)$$

  at any integer gives a result divisible by $i!$
- In other words, $(X)_i$ is a null polynomial modulo $p^{\nu_p(i!)}$

Null Polynomials

- Evaluating the **falling factorial polynomial**

$$(X)_i = X(X-1)\cdot\ldots\cdot(X-i+1)$$

  at any integer gives a result divisible by $i!$
- In other words, $(X)_i$ is a null polynomial modulo $p^{\nu_p(i!)}$
- The set of all null polynomials $\mathcal{O}_{p^e}$ includes
  - $(X)_i$ for $\nu_p(i!) \geq e$
  - $p^{e-\nu_p(i!)} \cdot (X)_i$ for $\nu_p(i!) < e$
  - Linear combinations of the above

Null Polynomials

- Evaluating the **falling factorial polynomial**

$$(X)_i = X(X-1)\cdot\ldots\cdot(X-i+1)$$

  at any integer gives a result divisible by $i!$
- In other words, $(X)_i$ is a null polynomial modulo $p^{\nu_p(i!)}$
- The set of all null polynomials $\mathcal{O}_{p^e}$ includes
  - $(X)_i$ for $\nu_p(i!) \geq e$
  - $p^{e-\nu_p(i!)} \cdot (X)_i$ for $\nu_p(i!) < e$
  - Linear combinations of the above
- Let $(X)_{\mu(p^e)}$ be the monic null polynomial of lowest degree
  - The relation $\mu(p^e) \leq p \cdot e$ always holds

Lowest Degree Representation

- Starting from any representation $F(X)$ of a polyfunction f, the Euclidean division
$$F(X) = (X)_{\mu(p^e)} \cdot Q(X) + F'(X)$$
gives another representation $F'(X)$ of degree $< \mu(p^e)$

Lowest Degree Representation

- Starting from any representation $F(X)$ of a polyfunction f, the Euclidean division
$$F(X) = (X)_{\mu(p^e)} \cdot Q(X) + F'(X)$$
gives another representation $F'(X)$ of degree $< \mu(p^e)$
- Every polyfunction has a representation of degree $< \mu(p^e) \leq p \cdot e$

Lowest Degree Representation

- Starting from any representation $F(X)$ of a polyfunction f, the Euclidean division
$$F(X) = (X)_{\mu(p^e)} \cdot Q(X) + F'(X)$$
gives another representation $F'(X)$ of degree $< \mu(p^e)$
- Every polyfunction has a representation of degree $< \mu(p^e) \leq p \cdot e$

Digit extraction function g_e

- Chen/Han representation $G_e^{CH}(X)$ has minimal degree $(p-1)\cdot(e-1)+1$
- Still we can search for representations that can be evaluated with even fewer scalar and non-scalar multiplications.

Improvement I: Parity

- Digit extraction is a **symmetric function**:
  - Even if $p = 2$: $g_e(-a) = g_e(a)$
  - Odd if $p > 2$: $g_e(-a) = -g_e(a)$

Improvement I: Parity

- Digit extraction is a **symmetric function**:
  - Even if $p = 2$: $g_e(-a) = g_e(a)$
  - Odd if $p > 2$: $g_e(-a) = -g_e(a)$
- We can choose a representation that contains either **only even- or odd-exponent** terms

Improvement I: Parity

- Digit extraction is a **symmetric function**:
  - Even if $p = 2$: $g_e(-a) = g_e(a)$
  - Odd if $p > 2$: $g_e(-a) = -g_e(a)$
- We can choose a representation that contains either **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

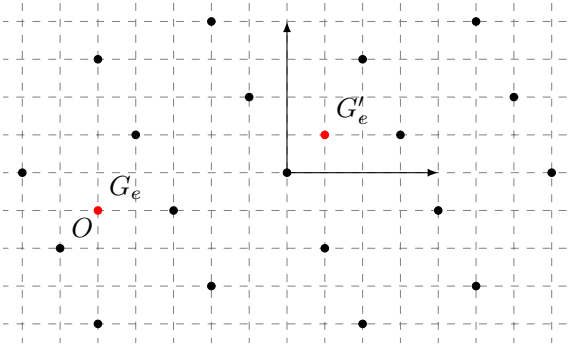
- Digit extraction is a **symmetric function**:
  - Even if $p = 2$: $g_e(-a) = g_e(a)$
  - Odd if $p > 2$: $g_e(-a) = -g_e(a)$
- We can choose a representation that contains either **only even- or odd-exponent** terms
  - For $p > 2$:

$$G_e(X) = (G_e^{CH}(X) - G_e^{CH}(-X))/2$$

  - 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**: $G'_e(X) = G_e(X) - O(X)$



Example

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

- Recall that Chen and Han find the following 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

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

- Resulting polynomial has **fewer terms** and **smaller coefficients**
- More efficient homomorphic evaluation and less noise growth

Improvement III: Function Composition

Idea: Decomposing the digit extraction function modulo p^e as

$$g_e = g_{e,e'} \circ g_{e'}$$

for some $e' < e$.

- The relevant domain of $g_{e,e'}$ is $S \coloneqq \mathsf{Range}(g_{e'}) \subset \mathbb{Z}_{p^e}$

Improvement III: Function Composition

Idea: Decomposing the digit extraction function modulo p^e as

$$g_e = g_{e,e'} \circ g_{e'}$$

for some $e' < e$.

- The relevant domain of $g_{e,e'}$ is $S \coloneqq \mathsf{Range}(g_{e'}) \subset \mathbb{Z}_{p^e}$
- Using null polynomials over S, we obtain a representation of $G_{e,e'}(X)$ with degree bound $p \cdot \lceil e/e' \rceil$

Improvement III: Function Composition

Idea: Decomposing the digit extraction function modulo p^e as

$$g_e = g_{e,e'} \circ g_{e'}$$

for some $e' < e$.

- The relevant domain of $g_{e,e'}$ is $S := \mathsf{Range}(g_{e'}) \subset \mathbb{Z}_{p^e}$
- Using null polynomials over S, we obtain a representation of $G_{e,e'}(X)$ with degree bound $p \cdot \lceil e/e' \rceil$
- Compared to Chen/Han, we have following complexity gain:
  - Non-scalar multiplications: $\mathcal{O}(\sqrt{pe}) \Rightarrow \mathcal{O}(\sqrt{p}\sqrt[4]{e})$
  - Scalar multiplications: $\mathcal{O}(pe) \Rightarrow \mathcal{O}(p\sqrt{e})$

Improvement III: Function Composition

Idea: Decomposing the digit extraction function modulo p^e as

$$g_e = g_{e,e'} \circ g_{e'}$$

for some $e' < e$.

- The relevant domain of $g_{e,e'}$ is $S := \mathsf{Range}(g_{e'}) \subset \mathbb{Z}_{p^e}$
- Using null polynomials over S, we obtain a representation of $G_{e,e'}(X)$ with degree bound $p \cdot \lceil e/e' \rceil$
- Compared to Chen/Han, we have following complexity gain:
  - Non-scalar multiplications: $\mathcal{O}(\sqrt{pe}) \Rightarrow \mathcal{O}(\sqrt{p}\sqrt[4]{e})$
  - Scalar multiplications: $\mathcal{O}(pe) \Rightarrow \mathcal{O}(p\sqrt{e})$
- Multiplicative depth increases with roughly $\lceil \log_2 p \rceil$

Example

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

- Recall that improvement I and II result in

$$G_8(X) = 13X^8 - 12X^6 \pmod{2^{25}}$$

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

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

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

- Function composition was combined with lattice trick
- Resulting polynomial has **low degree** and remarkably **small coefficients**

The Digit Removal Procedure in Bootstrapping

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

$$w = \underbrace{\blacksquare \cdots \blacksquare \blacksquare \cdots \blacksquare \blacksquare}_{e \text{ digits}}$$

Goal of digit removal:

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

This requires

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

Controlling the Depth of Digit Removal

Example

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

Chen/Han

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

Our approach

- Only uses our optimized representations $G_e(X)$
- Reuse polynomial evaluations while keeping the **same depth** 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 / Our standard method / Function composition

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

Conclusion

- Speed up bootstrapping for BGV and BFV up to $2.6\times$
  - Digit extraction function g_e
    - Parity
    - Lattice
    - Function composition
  - Digit removal procedure
- 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!

Full paper at Eurocrypt 2023
eprint.iacr.org/2022/1364

jiayi.kang@esat.kuleuven.be
robin.geelen@esat.kuleuven.be