Jeongeun Park[1], Sergi Rovira[2]
[1] imec-COSIC, KU Leuven, Leuven, Belgium
[2] WiSeCom, UPF, Barcelona, Spain
FHE.org 2023, Tokyo, Japan, 26 Mar 2023

Efficient TFHE Bootstrapping in the Multiparty Setting

Motivation

- Handling multiple users securely and efficiently for privacy preserving protocol is important for real world applications.
- Multiparty homomorphic encryption (MPHE; a.k.a. threshold-multikey FHE) gives the best solution for it in terms of computation and communication complexity.
- State of the art FHE schemes such as BGV, FV and CKKS are already efficiently extended to their MPHE version, but there is no concrete such extension of TFHE.
- TFHE with multi-digit secret key (in the single key setting) can be naturally adapted to the multiparty setting since the addition of secret keys produces a valid secret key:

$$S = s_1 + s_2 + \cdots + s_k,$$

where $\|S\|_\infty = k$ and $\|s_i\|_\infty = 1$.

Our Contribution

- We compare the bootstrapping complexity of the two existing works [LMK+22,JP22] which construct TFHE with multi-digit secret key.
- We conclude that [JP22] is the fastest and we extend it to the MPHE version by designing an efficient global bootstrapping key generation algorithm.
- To do this, we introduce a novel algorithm called *homomorphic indicator* which outputs a unit vector where the desired element is an encryption of 1, 0 elsewhere.
- We provide our implementation result and compare the theoretical MPHE extension of [LMK+22] to show which scheme is better in what parameter sets.

Homomorphic Indicator

Algorithm 1 Homomorphic Indicator (Hom.Indicator)

Input: $\{\mathbf{C}_i\}_{i\in[m]}$, A^{new} and A^{old}.
Output: A^{old}.
  for $i \leftarrow 1$ to k **do**
    for $j \leftarrow 1$ to k **do**
      $\mathsf{A}^{new}[j] := \mathsf{CMUX}_{\boxtimes}(\mathbf{C}_i, \mathsf{A}^{old}[j], \mathsf{A}^{old}[j-1])$
    end for
    $\mathsf{A}^{new}[0] := \mathsf{A}^{old}[0] \boxtimes (1 - \mathbf{C}_i)$
    for $j \leftarrow 0$ to k **do**
      $\mathsf{A}^{old}[j] := \mathsf{A}^{new}[j]$
    end for
  end for

- The *internal product* of two RGSW ciphertexts $\mathbf{C}_1$ and $\mathbf{C}_2$ is defined as follows:

$$\boxtimes : \mathsf{RGSW} \times \mathsf{RGSW} \to \mathsf{RGSW},\ (\mathbf{C}_1, \mathbf{C}_2) \mapsto (\mathbf{C}_1 \boxdot \mathbf{c}_1, \ldots, \mathbf{C}_1 \boxdot \mathbf{c}_{2\ell}).$$

- We instantiate a CMUX gate as follows:

$$\mathsf{CMUX}_{\boxtimes}(\mathbf{C}, \mathbf{C}_0, \mathbf{C}_1) \leftarrow (\mathbf{C}_1 - \mathbf{C}_0) \boxtimes \mathbf{C} + \mathbf{C}_0.$$

Global Bootstrapping Key Generation

Algorithm 2 Global bootstrapping key generation

Input: $\{\mathsf{bsk}_i\}_{i\in[k]}$, A^{new} and A^{old}.
Output: $\hat{\mathsf{bsk}}$.
  for $t \leftarrow 0$ to $n-1$ **do**
    for $i \leftarrow 1$ to k **do**
      Parse $\mathbf{C}_{i,t} := \mathsf{bsk}_i[t]$
    end for
    $\mathsf{A} := \mathsf{Hom.Indicator}(\{\mathbf{C}_{i,t}\}_{i\in[k]}, \mathsf{A}^{new}, \mathsf{A}^{old})$
    $\hat{\mathsf{bsk}}[t] := [\mathsf{A}[1], \ldots, \mathsf{A}[k]]$
    Refresh A^{new} and A^{old}
  end for

A Toy Example

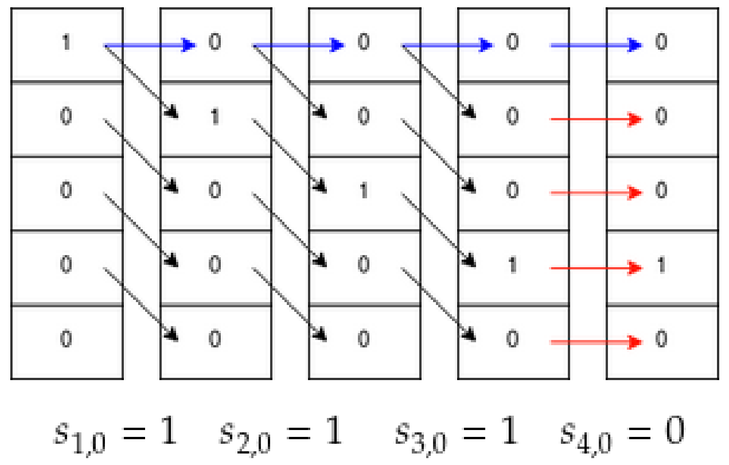

Figure 1: Example for k = 4 of our bootstarpping key generation. The blue arrow shows the direction of moving the first slot. The black arrows show that the elements in the new array comes from the previous slots in the old array when the corresponding secret key component is 1. The red arrow shows that the elements in the new array come from the same slots of the old array when the corresponding secret key component is 0.

Noise Analysis and Micro Benchmarks

Table 1: Comparison in terms of the number of expensive operations such as external products denoted by T_{mult} and a point-wise multiplication between two polynomials of degree N in FFT domain, denoted by T_{PM} used in blind rotation.

Scheme	Blind Rotation
[LMK+22]	$(1.5n + w) \cdot T_{mult}$
[JP22]	$n \cdot T_{mult} + k \cdot n \cdot (4 \cdot \ell \cdot T_{PM})$

- To know up to which k our MPHE approach based on [JP22] outperforms that of [LMK+22], we can use the above table and upper bound k as follows:

$$k \leq \frac{0.5 \cdot T_{mult}}{4 \cdot \ell \cdot T_{PM}} = \frac{1}{2} + \frac{(\ell+1)}{4\ell} \cdot \frac{T_{fft}}{T_{PM}}, \quad (1)$$

where T_{fft} denotes the time to convert a polynomial to the Fourier domain.
- To guarantee the correctness, we obtain the bound of k as follows:

$$k \leq \left(\frac{q}{96 \cdot \sqrt{\iota} \cdot \sqrt{n} \cdot \ell \cdot N \cdot g^2 \cdot \theta} \right)^{\frac{2}{3}}. \quad (2)$$

- Our experiments show a wide range for T_{PM}, ranging from 110 nanoseconds to 779 nanoseconds, while T_{fft} remains constant at an average of 7 microseconds.
- From these benchmarks, we can expect that our MPHE approach will be faster when k ranges from 3 to 47, for $N = 2^{11}$ and $\ell = 3$.
- For $N = 2^{15}$ (e.g., TFHE for large message spaces), the ratio between T_{fft} and T_{PM} will be larger and our approach will support more parties while keeping the same level of efficiency.

Experimental Results

Table 2: Parameter sets recommended, achieving at least 110-bit security based on LWE estimator for different number parties k. We indicate by $\log q$ and $\log Q$ the LWE and RLWE modulus, respectively. We define $\ell = O(\log q)$ and $\mathsf{B} = log(g)$ given the gadget vector $\mathbf{g} = (1, g, \ldots, g^{\ell-1})^t$. The values in the last three columns correspond to the average of 500 NAND operations, each performed with a freshly encrypted LWE ciphertext.

k	n	$\log q$	$\log Q$	B	ℓ	Time (seconds)	Bootstrapping noise	Bsk size (GB)
2	530	32	64	12	3	**0.20**	56.2 (24.2)	0.42
				6	8	0.48	**45.6 (13.6)**	0.45
4	495	32	64	11	4	**0.33**	56.12 (24.12)	0.63
				7	7	0.59	**48.97 (16.97)**	0.66
8	495	32	64	8	4	**0.46**	57.51 (57.51)	1.3
				7	6	0.70	**50.65 (18.65)**	1.1
16	495	32	64	10	5	**0.90**	58.37 (26.37)	2.3
				7	6	1.06	**52.79 (20.79)**	2.2

References

[LMK+22] Y. Lee, D. Micciancio, A. Kim, R. Choi, M. Deryabin, J. Eom, and D. Yoo, "Efficient FHEW Bootstrapping with Small Evaluation Keys, and Applications to Threshold Homomorphic Encryption," Eurocrypt 2022.

[JP22] M. Joye and P. Paillier, "Blind Rotation in Fully Homomorphic Encryption with Extended Keys"," CSCML 2022.

Acknowledgement

This work was partially supported by CyberSecurity Research Flanders under Grant VR20192203 and by the Spanish government, co-financed by the ESF.

Contact

- jeongeun.park@esat.kuleuven.be
- sergi.rovira@upf.edu