Plaintext slot permutations in batch ciphertexts for NTRU-based fully homomorphic encryption schemes

Bogdan Kulynych

Abstract

Batching using CRT allows to naturally parallelize computation for fully homomorphic encryption schemes over the integers or polynomial rings. This technique has been widely used in literature to improve performance of homomorphic algorithms [8] [3], but not any algorithm can be computed over batch ciphertext that support only support addition and multiplication isomorphically. Additional ability to permute plaintext slots in a batch ciphertext means that any binary circuit can be expressed as an algorithm over batch ciphertexts, leading to homomorphic computation overhead polylogarithmic in the security parameter. Plaintext slot permutations implemented as Galois group actions have only been initially proposed and explicitly described for the BGV encryption scheme. This work considers applying the Galois field-based machinery for plaintext slots permutation to NTRU and YASHE fully homomorphic encryption schemes. We also pair it with double-CRT representation, that, combined, allows for highly efficient polylog overhead homomorphic computation with popular NTRU-based FHE schemes.

Міністерство освіти і науки України
Національний Університет «Києво-Могилянська Академія»
Кафедра математики факультету інформатики

Гомоморфна криптографія. Перестановки слотів у групованих шифротекстах

Курсова робота за спеціальністю «Прикладна математика»

Керівник курсової роботи
Олійник Б.В.

Виконав студент МП-1, «Прикладна Математика»
Кулинич Б.В.

Київ – 2016

Contents

Introduction

Since the introduction of the first FHE scheme in the seminal work by Gentry [6], many new Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE) designs and implementations aimed at making homomorphic encryption practical for real-world applications appeared. This work focuses on schemes from NTRU family: namely, construction proposed by Doröz et al. [5] and variant of the original homomorphic construction by López-Alt et al. [9] called YASHE [1]. We consider batch ciphertext permutation techniques that are originally optimizations for the Brakerski-Gentry-Vaikuntanathan (BGV) scheme [2].

Batching of ciphertexts, that is, ability to simultaneously execute homomorphic operations on multiple encrypted values in SIMD ("single instruction multiple data") fashion, is one of the many optimizations that have been proposed to different FHE schemes. It was first introduced in [11] in the context of Gentry's original scheme, and the technique was later adapted to integer-based FHE schemes [3], BGV schemes [7], and recently, NTRU [10]. Notably, it was also shown in [7] that it is possible to permute plaintext slots in a batch ciphertext. Such operation allows to perform homomorphic computation with overhead that is polylogarithmic in the security parameter by expressing the computation circuit in terms of batch additions, multiplications, and permutations.

Even though the techniques from [7] are trivial to apply to schemes in algebraic setting similar to BGV, the batch permutation for NTRU is missing in the literature. It is not even mentioned in "Homomorphic AES evaluation using NTRU" [5], which, as the name hints, follows "Homomorphic AES evaluation" [8] where batch permutations in BGV ciphertexts are extensively used to optimize AES computation. This work considers plaintext slots permutation technique applied to the NTRU and YASHE schemes, i.e. we verify that Galois group actions permute plaintext slots in batch ciphertexts in NTRU setting as well.

Preliminaries

For all schemes in this work we use rings of dimension m defined by m-th cyclotomic polynomials $\Phi_m(X)$, $\mathbb{A} = \mathbb{Z}[X]/\Phi_m(X)$. We let $\mathbb{A}_q$ denote the set of elements of this ring reduced modulo q. The ring $\mathbb{A}$ is the ring of integers of

the m-th cyclotomic number field $\mathbb{K} = \mathbb{Q}(\zeta_m)$, where ζ_m is primitive m-th root of unity. We let $[a]_q$ for an element $a \in \mathbb{A}$ denote the reduction of a modulo q, fixing a unique representative in $(-q/2, q/2]$ for its equivalence class. We fix a set of L primes $p_0, p_1, ..., p_{L-1}$, and define t-th ciphertext modulus q_t to be $q_t = \prod_{i=0}^{L-t-1} p_i$, obtaining a decreasing moduli chain $q_0 > q_1 > ... > q_{L-1}$. The modulus of the fresh ciphertext is $q_0 = p_0 \cdot p_1 \cdot ... \cdot p_{L-1}$, and it decreases down to $q_{L-1} = p_0$ as more homomorphic operations are evaluated.

Let $\lceil \cdot \rfloor$ denote rounding to the nearest integer, $\lfloor \cdot \rfloor$ rounding down (floor). We define scaling factor Δ_q as $\Delta_q = \lfloor \frac{p}{q} \rfloor$.

Plaintext slots

We use p to denote the plaintext space modulus, thus the plaintexts are elements of $\mathbb{A}_p$. We assume that p does not divide m. As a result, polynomial $\Phi_m(X)$ factors modulo p into l irreducible factors, $\Phi_m(X) = F_0(X) \times F_1(X) ... \times F_{l-1}(X) \mod p$. Since $\mathbb{K}$ is Galois, all of the factors have the same degree $d = \phi(m)/l$. Thus, the plaintext space splits in the product of l finite fields:

$$\mathbb{A}_p \cong \mathbb{Z}[X]_p / F_0(X) \times \mathbb{Z}_p[X]/F_1(X) \times ... \times \mathbb{Z}_p[X]/F_{l-1}(X)$$

Each factor corresponds to a so-called plaintext slot. We view a polynomial $a \in \mathbb{A}_p$ as representing an l-dimensional vector $(a \mod F_i)_{i=0}^{l-1}$, or, equivalently, l-vector of elements in $\mathbb{F}_{p^d}$. Chinese remainder theorem is used to pack the vector into a single value in $\mathbb{A}_p$. We call this ciphertext representation *batch*, or CRT representation. We denote procedure of packing using CRT as $\mathsf{CRT} : \mathbb{F}_{p^d}^l \to \mathbb{A}_p$, and similarly, unpacking as $\mathsf{CRT}^{-1} : \mathbb{A}_p \to \mathbb{F}_{p^d}^l$.

Sampling from $\mathbb{Z}_q[X]/\Phi_m(X)$

Let us define following probability distributions:

- Gaussian $\mathcal{DG}_q(\sigma^2)$. Draw $a \in \mathbb{A}_q$ with coefficients from zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$ rounded to the nearest integer.
- Small polynomial $\mathcal{HWT}(h)$. For $h < \phi(m)$, draw $a \in \mathbb{A}$ with coefficients uniformly drawn from $\{0, +1, -1\}$, such that a has exactly h non-zero entries.

We will denote as $x \leftarrow \mathcal{D}$ drawing x from the distribution $\mathcal{D}$.

Partial scheme definitions

We use variants and notation close to a single-framework review by Costache and Smart [4]. For the sake of conciseness, we only provide partial scheme definitions consisting of key generation, encryption, and decryption procedures, ommiting other important components and details that are irrelevant in the scope of this work.

The following are procedures for NTRU and YASHE encryption schemes.

$\mathsf{KeyGen}(h)$. Sample $f, g \leftarrow \mathcal{HWT}(h)$. If f is not invertible in A_{q_0}, resample f. Otherwise, set private key $\mathfrak{sk} = p \cdot f + 1$, and public key $\mathfrak{pk} = [p \cdot g \cdot f^{-1}]_{q_0}$. Output $(\mathfrak{sk}, \mathfrak{pk})$.

$\mathsf{NTRU.Encrypt}(\mathfrak{pk}, m \in \mathbb{A}_p)$. Sample $e_1, e_2 \leftarrow \mathcal{DG}_{q_0}(\sigma^2)$. Encrypt m:

$$c = [e_1 \cdot \mathfrak{pk} + p \cdot e_2 + m]_{q_0}$$

Set the ciphertext level $t = 0$. Output $\mathfrak{c} = (c, 0)$.

$\mathsf{NTRU.Decrypt}(\mathfrak{sk}, \mathfrak{c} \in \mathbb{A}_{q_t})$. Output decrypted message:

$$m' = \Big[[\mathfrak{sk} \cdot c]_{q_t}\Big]_p$$

$\mathsf{YASHE.Encrypt}(\mathfrak{pk}, m \in \mathbb{A}_p)$. Sample $e_1, e_2 \leftarrow \mathcal{DG}_{q_0}(\sigma^2)$. Output:

$$c = [e_1 \cdot \mathfrak{pk} + \Delta_{q_0} \cdot e_2 + m]_{q_0}$$

Set the ciphertext level $t = 0$. Output $\mathfrak{c} = (c, 0)$.

$\mathsf{YASHE.Decrypt}(\mathfrak{sk}, \mathfrak{c} \in \mathbb{A}_{q_t})$. Output decrypted message:

$$m' = \Big[\Big\lceil \frac{p}{q_t} [\mathfrak{sk} \cdot c]_{q_t} \Big\rfloor\Big]_p$$

Permutations of plaintext slots

An algorithm that allows to perform arbitrary permutation of the plaintext slots using just the homomorphic Select operation and cyclic rotations of plaintext slots is shown in [7]. We verify that a method of performing cyclic rotations as automorphism of $\mathbb{K}$ can be easily used in NTRU setting.

Recall that Galois group $\mathcal{Gal}(\mathbb{K}/\mathbb{Q})$ action is a result of applying transformation $\kappa_i : f(X) \mapsto f(X^i) \mod \Phi_m(X), q_t$ for $i \in \mathbb{Z}_m^*$. Importantly, for some values i and plaintext vector $\mathbf{a} = (a_0, a_1, ..., a_{l-1}) \in \mathbb{F}_{p^d}^l$ with corresponding

$f = \mathsf{CRT}(\mathbf{a})$ and its encryption c under $\mathfrak{sk}$, the transformation $c^{(i)} = \kappa_i(c)$ produces a ciphertext, that decrypts under $\kappa_i(\mathfrak{sk})$ to a polynomial whose plaintext slots are rotated values of $\mathbf{a}$: $\mathsf{CRT}^{-1}(f^{(i)}) = (a_k, ..., a_{l-1}, a_0, a_1, ..., a_{k-1})$ for some $k \in \{1, 2, ..., l-1\}$. Namely, κ_i rotates the plaintext slots when i is not in $\{p^j \mid j = 0, 1, ..., d-1\}$.

Automorphisms

We will look at the effect of the Galois group action on decryption.

NTRU If NTRU ciphertext $\mathfrak{c}$ is decryptable, we have over $\mathbb{Z}_{q_t}[X]/\Phi_m(X)$:

$$[\mathfrak{sk} \cdot c]_{q_t} = m + p \cdot (e'_1 \cdot g + e'_2 + f \cdot m) + p^2 \cdot e'_2 \cdot f$$

That is, for some $r, u, v \in \mathbb{A}_{q_t}$ the following equality in $\mathbb{Z}_{q_t}[X]$ holds:

$$\mathfrak{sk}(X) \cdot c(X) = m(X) + p \cdot u(X) + p^2 \cdot v(X) + r(X) \cdot \Phi_m(X)$$

If we apply κ_i to both sides of the equation, the equality will be preserved, since κ_i is an automorphism in $\mathbb{K}$:

$$\mathfrak{sk}(X^i) \cdot c(X^i) = m(X^i) + p \cdot u(X^i) + p^2 \cdot v(X^i) + r(X^i) \cdot \Phi_m(X^i)$$

It is easy to show that for $i \in Z_m^*$, $\Phi_m(X)$ divides $\Phi_m(X^i)$, thus, over the $\mathbb{Z}_{q_t}[X]/\Phi_m(X)$, ciphertext decrypts to $m(X^i)$ under $\mathfrak{sk}(X^i)$:

$$\mathfrak{sk}(X^i) \cdot c(X^i) = m(X^i) + p \cdot u(X^i) + p^2 \cdot v(X^i)$$

YASHE We apply the same reasoning to YASHE ciphertexts. Recall that $\Delta_{q_t} = \lfloor \frac{q_t}{p} \rfloor = \frac{q_t}{p} - \epsilon$ for some ϵ in $[0, 1)$. When YASHE ciphertext $\mathfrak{c}$ is decryptable, we have for some $e'_1, e'_2 \in \mathbb{A}_{q_t}$ over $\mathbb{Z}_{q_t}[X]/\Phi_m(X)$:

$$\frac{p}{q_t}\mathfrak{sk} \cdot c = \frac{p \cdot (\frac{q_t}{p} - \epsilon)}{q_t} + \frac{p \cdot e'_1}{q_t} + p \cdot e'_2 = m + p \cdot \frac{e'_1 - \epsilon \cdot m}{q_t} + p \cdot e'_2$$

In terms of polynomials, for $u, v, r \in \mathbb{A}_{q_t}$ we have over $\mathbb{Z}_{q_t}[X]$:

$$\frac{p}{q_t}\mathfrak{sk}(X) \cdot c(X) = m(X) + p \cdot u(X) + \frac{p}{q_t} \cdot v(X) + r(X) \cdot \Phi_m(X)$$

After applying, κ_i we have over $\mathbb{Z}_{q_t}[X]$:

$$\frac{p}{q_t}\mathfrak{sk}(X^i) \cdot c(X^i) = m(X^i) + p \cdot u(X^i) + \frac{p}{q_t} \cdot v(X^i) + r(X^i) \cdot \Phi_m(X^i)$$

This again decrypts to $m(X^i)$ under $\mathfrak{sk}(X^i)$ over $\mathbb{Z}_{q_t}[X]/\Phi_m(X)$:

$$\frac{p}{q_t}\mathfrak{sk}(X^i) \cdot c(X^i) = m(X^i) + p \cdot u(X^i) + \frac{p}{q_t} \cdot v(X^i)$$

Conclusion

Plaintext slot rotations from Galois group actions, and therefore plaintext slot permutations, can be used in NTRU-based homomorphic encryption schemes. This follows from the similarity of algebraic settings of the BGV as described in [7] and NTRU.

References

1. Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. In Martijn Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2013.

2. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

3. JungHee Cheon, Jean-Sébastien Coron, Jinsu Kim, MoonSung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch Fully Homomorphic Encryption over the Integers. In Thomas Johansson and PhongQ. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335. Springer Berlin Heidelberg, 2013.

4. Anamaria Costache and Nigel P Smart. Which ring based somewhat homomorphic encryption scheme is best?

5. Yarkın Doröz, Yin Hu, and Berk Sunar. Homomorphic aes evaluation using the modified ltv scheme. *Designs, Codes and Cryptography*, pages 1–26, 2015.

6. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

7. Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully Homomorphic Encryption with Polylog Overhead. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2012.

8. Craig Gentry, Shai Halevi, and Nigel P. Smart. Homomorphic Evaluation of the AES Circuit. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 850–867. Springer, 2012.

9. Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1219–1234. ACM, 2012.

10. Kurt Rohloff and David Bruce Cousins. A scalable implementation of fully homomorphic encryption built on ntru. In *Financial Cryptography and Data Security*, pages 221–234. Springer, 2014.

11. N.P. Smart and F. Vercauteren. Fully Homomorphic SIMD Operations. Cryptology ePrint Archive, Report 2011/133, 2011. http://eprint.iacr.org/.