


## A BRIEF SUMMARY OF THE FULL RNS VARIANT OF SOME HOMOMORPHIC ENCRYPTIONS

 Nargiz Khankishiyeva Hati\*

Ege University, Izmir, Türkiye

---

**Abstract.** Since discovery of fully homomorphic encryption by Gentry in 2009, interest of lattice-based cryptography has significantly increased. Several researches have improved the efficiency of homomorphic encryption (HE) schemes. Despite all those powerful results, huge computational cost of underlying operations limits feasibility of practical implementations. In order to avoid computations over large numbers, residue number system arithmetic was included into HE cryptosystems scheme. In this paper, residue number system (RNS) and its representation were introduced and applications of RNS representation in some homomorphic encryptions was presented and some of them are analyzed extensively. This paper aims to present a brief summary of RNS variants of some homomorphic encryptions.

---

**Keywords:** Residue Number System, Chinese Remainder Theorem, Homomorphic Encryptions, Approximate Arithmetic.

\***Corresponding author:** Nargiz Khankishiyeva Hati, Ege University, Izmir, Türkiye,

e-mail: [nargizkhankishiyeva@gmail.com](mailto:nargizkhankishiyeva@gmail.com)

*Received: 3 April 2023; Revised: 22 June 2023; Accepted: 10 July 2023; Published: 31 August 2023.*

---

## 1 Introduction

In cryptography, there are many computations over huge numbers, but residue number system (RNS) is supplied with efficient arithmetic. That is why, cryptography is interested in RNS's implementations on some cryptosystems. There are some cryptosystems based on RNS (Chervjakov et al., 2015).

Homomorphic encryptions are able to do computations on encrypted data without decryption. Gentry (2009) offered fully homomorphic encryptions (FHE) based on ideal lattices. FHE is divided into two categories (Babenko & Trepacheva, 2019). First category includes cryptosystems based on Gentry's method. Second category includes cryptosystems based on different mathematical objects, like matrix, polynomial based and finally, RNS based cryptosystems.

The technology of homomorphic encryptions has improved rapidly in a few years. Various HE schemes was offered and have become more practical. The RNS implemented in some cryptosystems such as RSA, ECC (Elliptic Curve Cryptography), pairings etc., more recently, it has been used to accelerate fully homomorphic encryptions as lattice-based cryptography (Bajard et al., 2015). These implementations focused on arithmetic operations in RNS, using Chinese Remainder Theorem (CRT) to represent and manipulate the large coefficients in the ciphertext polynomials.

## 2 The Residue Number System

The RNS is considered an alternative to a binary representation of numbers. A number is represented as a set of remainders (residues) of dividing that number by some moduli. Let's

take a natural number  $m$  and call it module. The remainders of division of an arbitrary number by  $m = 6$ , are  $\{0,1,2,3,4,5\}$ . It can be collected all natural numbers under the six groups according to the remainder obtained by dividing of all natural numbers by 6:

$$\begin{aligned} a_n &= 6n - 5, \\ a_n &= 6n - 4, \\ a_n &= 6n - 3, \\ a_n &= 6n - 2, \\ a_n &= 6n - 1, \\ a_n &= 6(n - 1), n \in N^+ \end{aligned}$$

For the module  $m$ , there is  $m$  residue classes. Because remainders of the division of all natural numbers by  $m$  are the elements of the set  $\{0, 1, \dots, m - 2, m - 1\}$ .

In residue number system, decimal numbers  $A_{10}$  are shown like below:

$$A_{10} = (a_1 a_2 \dots a_k)_{RNS}$$

Here,

$$a_i = A - \left\lfloor \frac{A}{p_i} \right\rfloor \cdot p_i, i = 1, 2, \dots, k.$$

$\left\lfloor \frac{A}{p_i} \right\rfloor$  shows the whole part of the division,  $p_i$  shows the different bases and are prime numbers.

## 2.1 The RNS Representation

Let  $\mathcal{B} = \{p_0, \dots, p_{k-1}\}$  be a basis and let  $P = \prod_{i=0}^{k-1} p_i$ . We represent by  $[\cdot]_{\mathcal{B}}$  the map from  $\mathbb{Z}_P$  to  $\prod_{i=0}^{k-1} \mathbb{Z}_{p_i}$ , defined by  $a \mapsto [a]_{\mathcal{B}} = ([a]_{p_i})_{0 \leq i < k}$ . It is a ring isomorphism from the CRT and  $[a]_{\mathcal{B}}$  is called the residue number system (RNS) representation of  $a \in \mathbb{Z}_P$ . The reverse representation is the important part of arithmetic in RNS (Isupov, 2021). The classical method of searching for the whole value of the number by residue numbers is got from constructive proof of CRT (Knuth, 1997).

$$\begin{aligned} a &= \left( \sum_{i=0}^{k-1} [a^{(i)} \cdot \hat{p}_i^{-1}]_p \cdot \hat{p}_j \right)_P \\ \hat{p}_i &= \prod_{i' \neq i} p_{i'} \in \mathbb{Z}. \end{aligned}$$

## 2.2 Fast Basis Conversion

For a basis  $\{p_0, \dots, p_{k-1}, q_0, \dots, q_{l-1}\}$ ,  $\mathcal{B} = \{p_0, \dots, p_{k-1}\}$  and  $C = \{q_0, \dots, q_{l-1}\}$  are its subbases. Their product will be denoted by, respectively  $P = \prod_{i=0}^{k-1} p_i$  and  $Q = \prod_{j=0}^{l-1} q_j$ . Then one can convert the RNS representation  $[a]_C = (a^{(0)}, \dots, a^{(l-1)}) \in \mathbb{Z}_{q_0} \dots \mathbb{Z}_{q_{l-1}}$  of an integer  $a \in \mathbb{Z}_Q$  into an element of  $\mathbb{Z}_{p_0} \dots \mathbb{Z}_{p_{k-1}}$  by computing

$$Conv_{C \rightarrow \mathcal{B}}([a]_C) = \left( \sum_{j=0}^{l-1} [a^{(j)} \cdot \hat{q}_j^{-1}]_{q_j} \cdot \hat{q}_j \pmod{p_i} \right)_{0 \leq i < k},$$

where  $\hat{q}_j = \prod_{j' \neq j} q_{j'} \in \mathbb{Z}$ .  $\sum_{j=0}^{l-1} [a^{(j)} \cdot \hat{q}_j^{-1}]_{q_j} \cdot \hat{q}_j \pmod{p_i} = a + Q \cdot e$ . Here  $Conv_{C \rightarrow \mathcal{B}}([a]_C)$  is the RNS representation of  $a + Q \cdot e$  with respect to the basis  $\mathcal{B}$  (Cheon et al., 2018).

### 2.3 Approximate Modulus Raising

For the basis  $\mathcal{B}=\{p_0, \dots, p_{k-1}\}$ ,  $C=\{q_0, \dots, q_{l-1}\}$  and  $\mathcal{D}=\{p_0, \dots, p_{k-1}, q_0, \dots, q_{l-1}\}$ ,  
Algorithm 1.

- 1: procedure  $ModUp_{C \rightarrow D}(a^{(0)}, a^{(1)}, \dots, a^{(l-1)})$
- 2:      $(\tilde{a}^{(0)}, \dots, \tilde{a}^{(k-1)}) \leftarrow Conv_{C \rightarrow \mathcal{B}}([a]_C)$ .
- 3: return  $(\tilde{a}^{(0)}, \dots, \tilde{a}^{(k-1)}, a^{(0)}, \dots, a^{(l-1)})$
- 4: end procedure.

### 2.4 Approximate Modulus Reduction

The aim of approximate modulus reduction is reduced to a problem of finding small  $\tilde{a} = \tilde{b} - P \cdot b$  satisfying  $\tilde{a} \equiv \tilde{b} \pmod{P}$ .

Algorithm 2.

- 1: procedure  $ModDown_{D \rightarrow C}(\tilde{b}^{(0)}, \dots, \tilde{b}^{(k+l-1)})$
- 2:      $(\tilde{a}^{(0)}, \dots, \tilde{a}^{(l-1)}) \leftarrow Conv_{B \rightarrow C}(\tilde{b}^{(0)}, \dots, \tilde{b}^{(k-1)})$
- 3:     for  $0 \leq j < l$  do
- 4:          $b^{(j)} = (\prod_{i=0}^{k-1} p_i)^{-1} \cdot (\tilde{b}^{(k+j)} - \tilde{a}^j) \pmod{q_j}$ .
- 5:     end for
- 6: return  $(b^{(0)}, \dots, b^{(l-1)})$ .
- 7: end procedure

## 3 RNS in Cryptosystems

There are many implementations of RNS in encryption schemes. Some of them are analyzed through the paper.

### 3.1 A Full RNS Representation of RSA

There are many variants of RNS implementation of RSA (Fadulilahi et al., 2015; Jain, 2017; Manochehri et al., 2010; Nozako et al., 2001). First RNS implementation of RSA is presented by Impert and Bajard (2004), which doesn't require any conversion. The message is directly considered as a value represented in RNS and all operations is computed within the system.

### 3.2 A Full RNS Representation of FV like Somewhat HE Schemes

The RNS implementation of FV focused on CRT representation applied to the large coefficients, proposed by Bajard et al. (2016). This work offers a way to reduce the need for multi-precision arithmetic and suggests techniques to enable a full RNS variant of the FV-like schemes (Fan & Vercauteren, 2012).

### 3.3 A Full RNS Representation Variant of the BGV (Brakerski-Gentry-Vaikuntanathan)

BGV is a leveled FHE scheme which introduces a new parameter  $L$ . BGV algorithm introduces several additional parameters and functions beyond Basic LWE and RLWE algorithms (Brakerski et al., 2012). Gentry, Halevi and Smart (2015) presented in their work a different variant of the Brakerski-Vaikuntanathan key-switching technique that doesn't require eliminating the norm of the ciphertext vector, and a method of applying the Brakerski-Gentry-Vaikuntanathan modulus switching transformation on ciphertexts in CRT representation.

Implementing RNS in BFV (Brakerski-Fan-Vercauteren) seems harder than BGV. One difference is that in BGV numbers are typically scaled by just single-precision factors, while in

BFV these factors are often big, of order similar to the multi-precision modulus  $q$ . Another difference is that features more of these scaling operations than BGV (Halevi et al., 2018).

### 3.4 A Full RNS Representation of the BFV

Plaintext space in BFV is  $\mathbb{Z}_t$  for some modulus  $t > 1$ , where secret keys and ciphertexts are dimension- $n$  vectors over  $\mathbb{Z}_q^n$  for some other modulus  $q \gg t$  (Halevi et al., 2018).

*KeyGen.* For the secret key,  $s \leftarrow \chi_{key}$  and  $sk : - (1, s) \in R^2$ . For the public encryption key set  $(a^{(0)}, \dots, a^{(L)}) \leftarrow U(\prod_{j=0}^L R_{q_j})$  and  $e \leftarrow \chi_{err}$ . Set the public key as

$$pk \leftarrow \left( pk^{(j)} = \left( b^{(j)}, a^{(j)} \right) \right) \in R_{q_j}^2, \quad 0 \leq j \leq L.$$

where  $b^{(j)} \leftarrow -a^{(j)} \cdot s + e \pmod{q_j}$  for  $0 \leq j \leq L$ .

For relinearization, set a uniform  $\alpha_j \in R_q$  and  $e_j \leftarrow \chi_{err}$  and  $\beta_j = \left[ q_j q_j^* s^2 - \alpha_j s + e_j \right]_q$  for each  $j = 1, 2, \dots, k$ .

*Enc<sub>pk</sub>(m).* For  $m \in R_t$ , sample  $v \leftarrow \chi_{enc}$  and  $e_0, e_1 \leftarrow \chi_{err}$ . Output the ciphertext  $ct = (ct^{(j)})_{0 \leq j \leq L} \in \prod_{j=0}^L R_{q_j}^2$ , where  $ct^{(j)} \leftarrow v \cdot pk^{(j)} + (e_0, e_1) + (\Delta m, 0) \pmod{q_j}$  for  $0 \leq j \leq L$ , where  $\Delta = \frac{q}{t}$ .

*Dec<sub>sk</sub>(ct).* For  $ct = (ct^{(j)} = (c_0^{(j)}, c_1^{(j)}))_{0 \leq j \leq L}$ , output  $\langle ct^{(0)}, sk \rangle \pmod{q_0} = \left[ c_0^{(0)} + c_1^{(0)} s \right]_{q_0} = x$ ,  $m := \lceil \lceil x \cdot t / q \rceil \rceil_t$ .

*Add(ct, ct').*  $ct = \left( ct^{(j)} = (c_0^{(j)}, c_1^{(j)}) \right)_{0 \leq j \leq l}$  and  $ct' = \left( ct'^{(j)} = (c_0'^{(j)}, c_1'^{(j)}) \right)_{0 \leq j \leq l}$  are given two ciphertexts.  $ct_{add} = \left( ct^{(j)} + ct'^{(j)} \right)_{0 \leq j < l} = (c_0^{(j)} + c_0'^{(j)}, c_1^{(j)} + c_1'^{(j)})_{0 \leq j < l}$ .

*Mult<sub>evk</sub>(ct, ct').*  $ct = \left( ct^{(j)} = (c_0^{(j)}, c_1^{(j)}) \right)_{0 \leq j \leq l}$  and  $ct' = \left( ct'^{(j)} = (c_0'^{(j)}, c_1'^{(j)}) \right)_{0 \leq j \leq l}$  are given two ciphertexts. Output  $ct_{mult} = (\tilde{c}_0^{(j)}, \tilde{c}_1^{(j)}) \in R_{q_j}^2$  for  $j = 0, \dots, l$ . The process begins by extending the CRT basis. This gives us a representation of each coefficient in  $c_y^x$ , in the larger ring  $\mathbb{Z}_{qp}$ , which in turns yields a representation of the  $c_y^x$ 's in the larger ring  $R_{qp}$ . Next we compute the three elements  $\tilde{c}_0^{(j+i)} = c_0^{(j+i)} c_0'^{(j+i)}$ ,  $\tilde{c}_1^{(j+i)} = c_0^{(j+i)} c_1'^{(j+i)} + c_1^{(j+i)} c_0'^{(j+i)}$ ,  $\tilde{c}_2^{(j+i)} = c_1^{(j+i)} c_1'^{(j+i)}$ , for  $j = 0, \dots, l-1$  and  $i = 0, \dots, k-1$ , where all the operations are in the ring  $R_{qp}$ . Then the procedure of scaling back to  $R_q$  begins and it gives the power-basis representation of the elements  $c_m^{*(j)} = \left[ \lceil \lceil t / q \rceil \tilde{c}_m^{(j)} \rceil \right]_q \in R_q$  for  $m = 0, 1, 2$  and  $j = 0, \dots, l$ . For relinearization for each  $q_j$ ,  $\tilde{c}_0^{(j)} = \left[ \sum_{i=0}^l [\beta_i]_{q_j} \cdot c_2^{*(i)} \right]_{q_j}$  and  $\tilde{c}_1^{(j)} = \left[ \sum_{i=0}^l [\alpha_i]_{q_j} \cdot c_2^{*(i)} \right]_{q_j}$  and then  $\tilde{c}_0^{(j)} = \left[ c_0^{*(j)} + \tilde{c}_0^{(j)} \right]_{q_j}$  and  $\tilde{c}_1^{(j)} = \left[ c_1^{*(j)} + \tilde{c}_1^{(j)} \right]_{q_j}$ .

### 3.5 A Full RNS Representation Variant of the Approximate HE

HE for arithmetic of approximate numbers (HEAAN) supports an approximate addition and multiplication of encrypted data, together with a new rescaling procedure for managing the magnitude of plaintext. The name of the algorithm is changed by the authors to the CKKS algorithm (Ozdemir & Koc, 2022). The CKKS scheme works in the ring of polynomials with the integer coefficients modulo  $m$ th cyclotomic polynomial  $\phi_m(x)$  that is  $R = \mathbb{Z}[x]/(\phi_m(x))$  (Cheon et al., 2017).

The RNS variant of CKKS is introduced by Cheon et al. (2018) and later several newer and better RNS variants are presented (Bossuat et al., 2021; Chen et al., 2019; Han & Ki, 2020; Lee et al., 2020). In this paper, the RNS variant offered by Cheon et al. (2018) is presented.

*KSGen(s<sub>1</sub>, s<sub>2</sub>).* For given secret polynomials  $s_1, s_2 \in R$ , sample uniform elements  $(a'^{(0)}, \dots, a'^{(k+L)}) \leftarrow U(\prod_{i=0}^{k-1} R_{p_i} \times \prod_{j=0}^L R_{q_j})$  and an error  $e' \leftarrow \chi_{err}$ . Output the switching key  $swk$  as

$$(swk^{(0)} = (b^{(0)}, a^{(0)}), \dots, swk^{(k+L)} = (b^{(k+L)}, a^{(k+L)})) \in \prod_{i=0}^{k-1} R_{p_i}^2 \times \prod_{j=0}^L R_{q_j}^2$$

where  $b^{(i)} \leftarrow -a^{(i)} \cdot s_2 + e' \pmod{p_i}$  for  $0 \leq i < k$  and  $b^{(k+j)} \leftarrow -a^{(k+j)} \cdot s_2 + [P]_{q_j} \cdot s_1 + e' \pmod{q_j}$  for  $0 \leq j \leq L$ .

KeyGen.

1. Sample  $s \leftarrow \chi_{key}$  and set the secret key as  $sk \leftarrow (1, s)$ .
2. Sample  $(a^{(0)}, \dots, a^{(L)}) \leftarrow U(\prod_{j=0}^L R_{q_j})$  and  $e \leftarrow \chi_{err}$ . Set the public key as

$$pk \leftarrow (pk^{(j)} = (b^{(j)}, a^{(j)})) \in R_{q_j}^2, \quad 0 \leq j \leq L.$$

where  $b^{(j)} \leftarrow -a^{(j)} \cdot s + e \pmod{q_j}$  for  $0 \leq j \leq L$ .

3. Set the evaluation key as  $evk \leftarrow KSGen(s^2, s)$ .

$Enc_{pk}(m)$ . For  $m \in R$ , sample  $v \leftarrow \chi_{enc}$  and  $e_0, e_1 \leftarrow \chi_{err}$ . Output the ciphertext  $ct = (ct^{(j)})_{0 \leq j \leq L} \in \prod_{j=0}^L R_{q_j}^2$ , where  $ct^{(j)} \leftarrow v \cdot pk^{(j)} + (m + e_0, e_1) \pmod{q_j}$  for  $0 \leq j \leq L$ .

$Dec_{sk}(ct)$ . For  $ct = (ct^{(j)})_{0 \leq j \leq l}$ , output  $\langle ct^{(0)}, sk \rangle \pmod{q_0}$ .

$Add(ct, ct')$ .  $ct = (ct^{(j)})_{0 \leq j \leq l} \in \prod_{j=0}^l R_{q_j}^2$ ,  $ct' = (ct'^{(j)})_{0 \leq j \leq l} \in \prod_{j=0}^l R_{q_j}^2$  are given ciphertexts, output a ciphertext  $ct_{add} = (ct_{add}^{(j)})_{0 \leq j \leq l} = (ct^{(j)} + ct'^{(j)})_{0 \leq j \leq l} \pmod{q_j}$ .

$Mult_{evk}(ct, ct')$ .  $ct = (ct^{(j)} = (c_0^{(j)}, c_1^{(j)}))_{0 \leq j \leq l}$  and  $ct' = (ct'^{(j)} = (c_0'^{(j)}, c_1'^{(j)}))_{0 \leq j \leq l}$  are given two ciphertexts. Output  $ct_{mult} \in \prod_{j=0}^l R_{q_j}^2$ .

1. For  $0 \leq j \leq l$ , compute

$$d_0^{(j)} \leftarrow (c_0^{(j)} c_0'^{(j)}) \pmod{q_j},$$

$$d_1^{(j)} \leftarrow (c_0^{(j)} c_1'^{(j)} + c_1^{(j)} c_0'^{(j)}) \pmod{q_j},$$

$$d_2^{(j)} \leftarrow (c_1^{(j)} c_1'^{(j)}) \pmod{q_j}.$$

2. Compute  $ModUp_{C_l \leftarrow D_l}(d_2^{(0)}, \dots, d_2^{(l)}) = (\tilde{d}_2^{(0)}, \dots, \tilde{d}_2^{(k-1)}, d_2^{(0)}, \dots, d_2^{(l)})$ .

3. Compute

$$\tilde{ct} = (\tilde{ct}^{(0)} = (\tilde{c}_0^{(0)}, \tilde{c}_1^{(0)}), \dots, \tilde{ct}^{(k+l)} = (\tilde{c}_0^{(k+l)}, \tilde{c}_1^{(k+l)})) \in \prod_{i=0}^{k-1} R_{p_i}^2 \times \prod_{j=0}^l R_{q_j}^2$$

where  $\tilde{ct}^{(i)} = \tilde{d}_2^{(i)} \cdot evk^{(i)} \pmod{p_i}$  and  $\tilde{ct}^{(k+j)} = \tilde{d}_2^{(j)} \cdot evk^{(k+j)} \pmod{q_j}$  for  $0 \leq i < k, 0 \leq j \leq l$ .

1. Compute

$$(\hat{c}_0^{(0)}, \dots, \hat{c}_0^{(l)}) \leftarrow ModDown_{D_l \rightarrow C_l}(\tilde{c}_0^{(0)}, \dots, \tilde{c}_0^{(k+l)}),$$

$$(\hat{c}_1^{(0)}, \dots, \hat{c}_1^{(l)}) \leftarrow ModDown_{D_l \rightarrow C_l}(\tilde{c}_1^{(0)}, \dots, \tilde{c}_1^{(k+l)}).$$

2. Output the ciphertext  $ct_{mult} = (ct_{mult}^{(j)})_{0 \leq j \leq l}$  where  $ct_{mult}^{(j)} \leftarrow (\hat{c}_0^{(j)} + d_0^{(j)}, \hat{c}_1^{(j)} + d_1^{(j)}) \pmod{q_j}$  for  $0 \leq j \leq l$ .  
 $RS(ct)$ .

For a level  $l$  ciphertext  $ct = (ct^{(j)} = (c_0^{(j)}, c_1^{(j)}))$  for  $0 \leq j \leq l$  and  $ct \in \prod_{j=0}^l R_{q_j}^2$ . Compute  $c_i^{\prime(j)} \leftarrow q_l^{-1} \cdot (c_i^{(j)} - c_i^{(l)}) \pmod{q_j}$  for  $i = 0, 1$  and for  $0 \leq j < l$ . Output the ciphertext  $ct' \leftarrow (ct^{\prime(j)} = (c_0^{\prime(j)}, c_1^{\prime(j)}))_{0 \leq j \leq l-1} \in \prod_{j=0}^{l-1} R_{q_j}^2$ .  
 For detailed information see Cheon et al. (2018).

## 4 Conclusion

This paper provided a brief look over RNS representation of some homomorphic operations. These implementations express message as number in RNS and using reverse formula of conversion from RNS into radix base, performs base extensions and base reductions. Thus, RNS gives a chance to work with small numbers and it concludes faster homomorphic operations in homomorphic encryptions. For example, the performance of basic operations was improved by the RNS implementation approximately 10 times compared to the original CKKS (Cheon et al., 2018). The newer RNS variants of CKKS has better performances. As the interest in cryptography continues, new methods will always be developed and better variants will emerge.

## References

- Babenko, L.K., Trepacheva, A.V., (2019). On the instability of two symmetric homomorphic cryptosystems based on the system of residual classes. *TR SPII RAN*, 1, 230-262 (In Russian).
- Bajard, J.C., Eynard, J., Merkiche, N., & Plantard, T., (2015). RNS Arithmetic Approach in Lattice-Based Cryptography: Accelerating the "Rounding-o-" Core Procedure. *In 22nd IEEE Symposium on Computer Arithmetic, ARITH 2015*, Lyon, France, June 22-24, 2015, pages 113-120. IEEE.
- Bajard, J.C., Eynard, J., Hasan, M.A., & Zucca, V., (2016). A full RNS variant of FV like somewhat homomorphic encryption schemes. *In International Conference on Selected Areas in Cryptography*, Springer.
- Bossuat, J.-P., Mouchet, C., Troncoso-Pastoriza, J., & Hubaux, J.-P., (2021). Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys. In A. Canteaut and F.-X. Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, Springer, LNCS Nr. 12696.
- Brakerski, Z., Gentry, C., & Vaikuntanathan, V., (2012), (leveled) fully homomorphic encryption without bootstrapping. In: *ITCS '12*, 309-325.
- Chen, H., Chillotti, I., & Song Y., (2019). Improved bootstrapping for approximate homomorphic encryption. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT*, 34–54. Springer, LNCS Nr. 11477.
- Cheon, J.H., Han K., Kim A., Kim M. & Song Y., (2018). A Full RNS Variant of Approximate Homomorphic Encryption. *25th International Conference*, Calgary, AB Canada, August 15-17.
- Cheon, J.H., Kim A., Kim A., & Song Y., (2017). Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology {ASIACRYPT 2017}*, Springer.

- Chervjakov, N.I., Babenko, M.G., & Kucherov, N.N., (2015). Development of a homomorphic information encryption system based on a polynomial residual classes system. *Siberian Electronic Mathematical Reports. Proceedings of the VI International Youth School-Conference Theory and numerical methods for solving inverse and ill-posed problems*, 12, 33-41. Available at <http://semr.math.nsc.ru/v12/c1-283.pdf> (accessed: 02.12.2018). (In Russian).
- Fadulillahi, I.R., Bankas E. & Ansuura, J.B.A.K., (2015). Efficient Algorithm for RNS Implementation of RSA. *International Journal of Computer Applications*, 127(5), 14-19.
- Fan, J., Vercauteren, F., (2012). Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144.
- Gentry C., (2009). A Fully Homomorphic Encryption Scheme. PhD thesis, Stanford University, [crypto.stanford.edu/craig](http://crypto.stanford.edu/craig).
- Gentry, C., Halevi, S., & Smart, N., (2012). Homomorphic evaluation of the AES circuit. In: *CRYPTO 2012*. LNCS, 850-867.
- Halevi S., Polyakov Y. & Shoup V., (2018). An Improved RNS Variant of the BFV Homomorphic Encryption Scheme: *The Cryptographers' Track at the RSA Conference 2019*, San Francisco, CA, USA, March 4-8, Proceedings, 83-105.
- Han, K., Ki, D., (2020). Better bootstrapping for approximate homomorphic encryption. In S. Jarecki, editor, *Topics in Cryptology - CT-RSA 2020*, 364–390. Springer, LNCS Nr. 12006.
- Imbert, L., Bajard, J.C. (2004). A Full RNS Implementation of RSA: *IEEE Transactions On Computers*, 53(5).
- Isupov, K.S., (2021). High-performance computing using the system of residual classes, *Software systems: theory and application*. Second Edition, 12, 132-192. (In Russian).
- Jain, S., (2017). Modified RSA Cryptographic System with Two Public Keys and Chinese Remainder Theorem. *International Journal on Computer Science and Engineering*.
- Knuth, D.E., (1997). *The Art of Computer Programming. V. 2: Seminumerical Algorithms*, 3rd ed., Addison-Wesley, USA, 1997, ISBN 0201896842.
- Lee, Y., Lee, J.-W., Kim, Y.-S., & No J.-S., (2020). Near-optimal polynomial for modulus reduction using L2-norm for approximate homomorphic encryption. *IEEE Access*, 8, 144321–144330.
- Manochehri, K., Pourmozafari, S. & Sadeghian, B., (2010). Montgomery and RNS for RSA Hardware Implementation. *Computing and Informatics*, 29, 849-880.
- Nozako, H., Motoyama, M., Shimbo, A., & Kawamura, S. (2001). Implementation of RSA Algorithm Based on RNS Montgomery Multiplication. Part of the *Lecture Notes in Computer Science* book series (LNCS volume 2162).
- Ozdemir, F., Koc, C.K., (2022). Development of Cryptography since Shannon. IACR ePrint Report 2022/100, January 26.
- Szabo, N.S., Tanaka, R.I., (1967). *Residue arithmetic and its applications to computer technology*. New York: McGraw-Hill, based on the authors' Report on residue (modular) arithmetic survey.
- Vishnevskij, A.K., Knjazev, V.V., (2015). Complex application of homomorphic cryptographic transformations for solving systems of linear algebraic equations. *High technology*, 16(11), 28-35. (In Russian).