1-1-2015

Babai round-off CVP method in RNS: application to latice based cryptographic protocols

Jean-Claude Bajard
Sorbonne University, bajard@lirmm.fr

Julien Eynard
Sorbonne University

Nabil Merkiche
DGA/MI, France

Thomas Plantard
University of Wollongong, thomaspl@uow.edu.au

Babai round-off CVP method in RNS: application to latice based cryptographic protocols

Abstract

Lattice based cryptography is claimed as a serious candidate for post quantum cryptography, it recently became an essential tool of modern cryptography. Nevertheless, if lattice based cryptography has made theoretical progresses, its chances to be adopted in practice are still low due to the cost of the computation. If some approaches like RSA and ECC have been strongly optimized - in particular their core arithmetic operations, the modular multiplication and/or the modular exponentiation - lattice based cryptography has not been arithmetically improved. This paper proposes to fill the gap with a new approach using Residue Number Systems, RNS, for one of the core arithmetic operation of lattice based cryptography: namely solving the Closest Vector Problem (CVP).

Disciplines

Engineering | Science and Technology Studies

Publication Details

Bajard, J., Eynard, J., Merkiche, N. & Plantard, T. (2015). Babai round-off CVP method in RNS application to latice based cryptographic protocols. Proceedings of the 14th International Symposium on Integrated Circuits (ISIC) (pp. 440-443). United States: IEEE.

Babaï Round-Off CVP method in RNS

Application to Lattice based cryptographic protocols

Jean-Claude Bajard, Julien Eynard[†]
Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France.
CNRS, UMR 7606, LIP6, F-75005, Paris, France. {jean-claude.bajard,julien.eynard}@lip6.fr

Nabil Merkiche
DGA/MI, Rennes, France.
nabil.merkiche@intradef.gouv.fr

Thomas Plantard
CCISR, Univ. of Wollongong, Australia.
thomaspl@uow.edu.au

***Abstract*—Lattice based cryptography is claimed as a serious candidate for post quantum cryptography, it recently became an essential tool of modern cryptography. Nevertheless, if lattice based cryptography has made theoretical progresses, its chances to be adopted in practice are still low due to the cost of the computation. If some approaches like RSA and ECC have been strongly optimized - in particular their core arithmetic operations, the modular multiplication and/or the modular exponentiation - lattice based cryptography has not been arithmetically improved. This paper proposes to fill the gap with a new approach using Residue Number Systems, RNS, for one of the core arithmetic operation of lattice based cryptography: namely solving the Closest Vector Problem (CVP).**

Keywords*—*Residue Number System; Base Conversions; Lattice; Closest Vector Problem

I. Introduction

The cryptography based on lattices appeared at the beginning of this century with initial propositions like GGH [12] and NTRU [14].

In few years, due to some properties of the lattices, some powerful cryptographic tools have been proposed for the first time: fully homomorphic encryption, multi-linear map and indistinguishability obfuscation [10]. Despite numerous attacks against the historical propositions, countermeasure after countermeasure, these systems are still available [8]. Even after numerous evolutions, they stay based on some simple proposals where the encryption is obtained by adding an ”error” to a vector of a lattice. This error represents the original message, and the vector obtained the ciphered one. All the security is based on the difficulty to reduce the public basis of the lattice in a Lovàsz reduced basis in which the Babaï algorithms can be performed [1].

Some recent approaches propose to use an oracle which gives some approximated closest vectors [16], [11], [20], and a Learning with Error method to find the closest vector. Nevertheless, they are not still sufficiently efficient in practice. Thus, an efficient computation of a closest vector remains a real challenge. As the Residue Number System (RNS) has been proved to be efficient for other cryptographic systems [13], [7], we suggest to study in this paper their use in lattice cryptography by implementing in RNS the Babaï Round-Off CVP method.

II. About Babaï Round-Off CVP

The main idea can be summarized in the following way. We create a lattice of full-rank ℓ from a strongly reduced basis given by a matrix G, and we construct another bad basis (in terms of lattice basis reduction) $H = UG$, U being a unimodular matrix. H can be in Hermite Normal Form [17].

The encryption mode [12] obeys the following scheme: $c = (m + kH)$ where m is the vector message composed of zeros and ones (or of small values with respect to the Lovàsz conditions), k is a vector such that $c = (c_1, 0, ..., 0)$ with c_1 huge, or $c = (c_1, c_2, ..., c_\ell)$, with small c_i's. The vector kH belongs to the lattice, and is a closest vector of c. In the following, we will consider that all the coefficients of c are positive, which is possible modulo a translation via a vector of the lattice. As the coefficients of m are small and G is strongly orthogonal, the message m is found using the Rounding Off algorithm of Babaï [1]. This operation is given by $m = c - \lfloor cG^{-1} \rceil \times G$, where $\lfloor cG^{-1} \rceil \times G$ represents the closest vector of the lattice. Since m is composed of small values, it is suggested to compute $c - \lfloor cG^{-1} \rceil \times G \bmod \beta$ where β is a small number, reducing by this way the complexity of the calculus. Nevertheless, though matrix G is an integer matrix, its inverse G^{-1} is not, i.e., is rational. The operation $\lfloor cG^{-1} \rceil$ must be done sufficiently precisely for obtaining a good rounding.

III. The RNS approach of the Rounding Off Babaï algorithm

In this work, we propose for this evaluation to use RNS systems which distribute the calculus on small values in a fully parallel way for additions and multiplications [24], [23]. These representations are based on the Chinese Remainder Theorem, a number α is represented by its residues $(\alpha_1, ..., \alpha_n)$ modulo a set of coprimes $(m_1, ..., m_n)$ called the RNS base. Hence, we are able to represent all the values from 0 to $M - 1 = \prod_{i=1}^{n} m_i - 1$.

[†] supported by Direction Générale de l'Armement (DGA).

978-1-4799-4833-8/14/$31.00 ©2014 IEEE

In this approach we use the modular reduction proposed by P. Montgomery [18] and adapted to RNS [19], [15], [2], both for the evaluation of $\lfloor cG^{-1} \rceil \times G$, and for the final reduction $\mathrm{mod} \beta$.

Our first purpose is to compute the value $\lfloor cG^{-1} \rceil$ in RNS. For this, we will transform this calculus in complete integer operation using that $G' = (\det G) \times G^{-1}$ is an integer matrix when G is one integer matrix. Thus we have: $\left\lfloor \frac{cG'}{\det G} \right\rceil = \lfloor cG^{-1} \rceil$.

In RNS, the division by $\det G$ is possible if it is an exact one and if $\det G$ is co-prime with the RNS Base. In this case we have,

$$\frac{cG' - (cG' \bmod \det G)}{\det G} = \left\lfloor \frac{cG'}{\det G} \right\rfloor .$$

As we want to compute $\left\lfloor \frac{cG'}{\det G} \right\rceil$, we will compute more precisely $\left\lfloor \frac{cG'}{\det G} + \frac{1}{2} v_1 \right\rfloor = \left\lfloor \frac{cG'}{\det G} \right\rceil$, where v_1 is the all-one vector (i.e. $v_1 = (1, 1, ..., 1)$).

If we develop this expression, we obtain:

$$\left\lfloor \frac{cG'}{\det G} \right\rceil = \left\lfloor \frac{cG'}{\det G} + \frac{1}{2} v_1 \right\rfloor$$

$$= \left\lfloor \frac{2cG' + \det G . v_1 - \left[(2cG' + \det G . v_1) \bmod (2 \det G) \right]}{2 \det G} \right\rfloor .$$

The most delicate operation is due to the modulo mod $(2 \det G)$, which requires in RNS a particular attention. The other operations can be directly implemented in RNS as is.

We note $\mathbf{D_G} = (\mathbf{2} \det \mathbf{G})$.

A. Evaluation of $[(2cG' + (\det G)v_1) \bmod D_G]$ in RNS

In this part, we consider the RNS bases $\mathcal{B}_1$ and $\mathcal{B}_2$ with $M_1 = \prod_{m \in \mathcal{B}_1} m$ and $M_2 = \prod_{m \in \mathcal{B}_2} m$. To compute a reduction of the form $a \bmod D_G$, the bases are selected such that $|a|_\infty < M_1 \times D_G$ and $2D_G < M_2$, assuming that D_G is coprime with the elements of $\mathcal{B}_1$ (which is generally the case, because $\det G$ is frequently a prime number).

The modular reduction can be done in RNS using the Montgomery algorithm [2]. The particularity of the approach is that the reduced value is obtained multiplied by a factor depending of the RNS base (in our case M_1^{-1}). When some values are fixed, G in our case, we can use precomputed values to avoid this extra final factor M_1^{-1}. Thus, we let denote by
$G" = 2G' \times M_1 \bmod D_G$
(recall that G^{-1} is not integer, but $G' = (\det G)G^{-1}$ is),
and $v" = (M_1 \times \det G)v_1 \bmod D_G$.

The "PreBabaïROffrns" has two modes, the *rns* one which gives the result on $\mathcal{B}_1$ and $\mathcal{B}_2$, and the one without option which gives the result modulo β adapted to a cryptographic context. This algorithm uses the Montgomery reduction in the states 1 and 3 of the procedure. The state 1 computes q_1 modulo M_1 such that $(a_2 + D_G \times q_2)$ gives a multiple of M_1, thus, in state 3, the division by M_1 is equivalent to a multiplication by its inverse. This last operation is possible in the base $\mathcal{B}_2$, since M_1 is coprime to M_2. Thus, base extensions are needed and correspond to states 2 and 4. Then, we obtain the value $r_2 \equiv [(2cG' + (\det G)v_1) \bmod D_G]$, with $|r_2|_\infty < 2D_G$, which is converted in $\mathcal{B}_1$ or modulo β with respect to the option.

Algorithm 1 PreBabaïROff_rns(*option*)

```
Input: $a = c \times G" + v"$, $a \in \mathbb{Z}^n$ given in the two bases $\mathcal{B}_1$ and $\mathcal{B}_2$, $|a|_\infty < M_1 \times D_G$, $2D_G < M_2$, all the values concerned by $G$ are considered as precomputed.
Output: $[(2cG' + (\det G)v_1) \bmod D_G]$ in $\mathcal{B}_1$ and $\mathcal{B}_2$ if (option = rns), else $\mathrm{mod}\beta$.
1: $q_1 \leftarrow (-D_G)^{-1} \times a_1$ in $\mathcal{B}_1$ (in other words, the evaluation is made modulo $M_1$),
2: $q_2 \leftarrow q_1$ Extension$_1$ from $\mathcal{B}_1$ to $\mathcal{B}_2$ of $q_1$,
3: $r_2 \leftarrow (a_2 + D_G \times q_2) \times M_1^{-1}$ in base $\mathcal{B}_2$,
   hence $r_2 \equiv (2cG' + (\det G)v_1) \bmod D_G$, with $|r_2|_\infty < 2D_G$
4: Extension$_2$ of $r_2$ in $\mathcal{B}_1$ if rns else modulo $\beta$.
```

B. Analysis of the first extension

For Extension$_1$ we need to extend q_1 exactly. A first solution could be to use an intermediate representation: Mixed Radix System [23]. However it is costly to compute, because the transformation from RNS to MRS involves lots of dependancies between intermediary results, which somehow breaks the parallelization provided by RNS. So we can replace steps 2 and 3 by an approach using extensions based on the Chinese Remainder Theorem (CRT). Such extension is performed as following: for a tuple of residues $(x \bmod m)_{m \in \mathcal{B}_1}$, then $x = \sum_{m \in \mathcal{B}_1} \left| (x \bmod m) \times \left| \frac{M_1}{m} \right|_m^{-1} \right|_m - \alpha M_1$, where $\alpha \in [0, |\mathcal{B}_1| - 1]$. α can be computed in different ways. In particular, it can be forgotten for first base extension [2] in algorithm 3. In this case, we propose a solution to reduce the final result in $[0, 2D_G)$ (like for classical Montgomery reduction) before applying second base extension. For that we use an extra modulo $\widehat{m}$ to recover α.

(We recall that $D_G = (2 \det G)$.)

Algorithm 2 Extension$_1$Bis

```
Input: $q_1$ in $\mathcal{B}_1$, $a_2$ in $\mathcal{B}_2$ and $a_{\widehat{m}} = a \bmod \widehat{m}$.
Output: $r'_2 \equiv aM_1^{-1}\widehat{m}^{-1} \bmod D_G$, $r'_2 < 2D_G$.
1: $q_2 \leftarrow \sum_{m \in \mathcal{B}_1} \left| q_{1,i} \left| \frac{M_1}{m_i} \right|_{m_i}^{-1} \right|_{m_i} \frac{M_1}{m_i}$ in $\mathcal{B}_2$,
   $q_{\widehat{m}} \leftarrow \sum_{m \in \mathcal{B}_1} \left| q_{1,i} \left| \frac{M_1}{m_i} \right|_{m_i}^{-1} \right|_{m_i} \frac{M_1}{m_i} \bmod \widehat{m}$
2: $r_2 \leftarrow (a_2 + D_G \times q_2) \times M_1^{-1}$ in $\mathcal{B}_2$ and
   $r_{\widehat{m}} \leftarrow (a_{\widehat{m}} + D_G \times q_{\widehat{m}}) \times M_1^{-1} \bmod \widehat{m}$,
3: $\widehat{q} \leftarrow (-D_G)^{-1} r_{\widehat{m}} \bmod \widehat{m}$
4: Extension of $\widehat{q}$ in $\mathcal{B}_2$ is just a duplication if $\widehat{m}$ is smaller than all the elements of $\mathcal{B}_2$
5: $r'_2 \leftarrow (r_2 + D_G \times \widehat{q}) \times \widehat{m}^{-1}$ in base $\mathcal{B}_2$
```

In step 1, $q_2 = q_1 + \alpha M_1$, thus

$$\begin{aligned} r_2 &= (a_2 + D_G \times q_2) \times M_1^{-1} \\ &= (a_2 + D_G \times (q_1 + \alpha M_1)) \times M_1^{-1} \\ &= (a_2 + D_G \times q_1) \times M_1^{-1} + \alpha D_G \\ &\equiv a\widehat{m}^{-1} \bmod D_G. \end{aligned}$$

Because size of base $\mathcal{B}_1$ has been chosen such that $a < M_1 \times D_G$, it is then clear that

$$\begin{aligned} (a + D_G \times q_1)/M_1 &= (a_2 + D_G \times q_2)/M_1^{-1} \bmod M_2 \\ &= r_2 < (2+\alpha)D_G. \end{aligned}$$

Hence we need to reduce it a second time. For that we use the extra modulo $\widehat{m}$ and we apply a second Montgomery reduction computing $\widehat{q}$, thus
$r_2' \equiv \left(a_2 \times M_1^{-1}\right) \times \widehat{m}^{-1} \bmod D_G$ with $r_2' < 2D_G$ when $\widehat{m} > |\mathcal{B}_1| + 1 \geq 2 + \alpha$.

We replace M_1 by $M_1' = M_1 \times \widehat{m}$. Hence, the precomputed values become
$G\text{”} = 2G' \times M_1' \bmod D_G$
and $v\text{”} = (M_1' \times \det G)v_1 \bmod D_G$.

C. Analysis of the second extension

For the second base extension, we can use an extra modulo $\widehat{m}$ with a Shenoy-Kumaresan approach [21]. But in this case, we cannot extract any information about the comparison of r_2' with D_G. Thus, we obtain $r_2' = (2cG' + (\det G)v_1) \bmod D_G$ or $[(2cG' + (\det G)v_1) \bmod D_G] + D_G$ which is still not satisfying for our purpose.

Hence, the second extension can be done in MRS which is a positional number system. In this case, during the conversion, a comparison with D_G is possible and if necessary we subtract D_G.

D. Complete "Round-Off" Closest Vector in RNS

Now, we come back to our problem which is to compute a closest vector with round-off formula: $\lfloor cG^{-1} \rceil \times G$. First we give a new version of the PreBabaïROff_rns including the new extension.

Algorithm 3 NewPreBabaïROff_rns(*option*)

```
Input: $a = c \times G\text{”} + v\text{”}$, $a \in \mathbb{Z}^n$ given in the bases $\mathcal{B}_1$, $\mathcal{B}_2$
    and $\widehat{m}$, $|a|_\infty < M_1 \times D_G$, $2D_G < M_2$, all the values
    concerned by $G$ are considered as precomputed.
Output: $[(2cG' + (\det G)v_1) \bmod D_G]$ in $\mathcal{B}_1$ and $\mathcal{B}_2$ if (option = rns), else $\bmod \beta$.
1: $q_1 \leftarrow (-D_G)^{-1} \times a_1$ in $\mathcal{B}_1$ (in other words, the evaluation
   is made modulo $M_1$),
2: $r_2' \leftarrow \mathrm{Extension_1Bis}(q_1, \mathcal{B}_1, \mathcal{B}_2, \widehat{m})$,
3: $\widetilde{r}_2 \leftarrow r_2'$ conversion in mixed radix,
4: Comparison of $\widetilde{r}_2$ with $(2 \det G)$,
5: Extension of $\widetilde{r}_2$ in $\mathcal{B}_1$ if rns else modulo $\beta$,
6: Subtraction of $D_G$ if necessary.
```

NewPreBabaïROff_rns algorithm gives $\lfloor cG^{-1} \rceil$ in bases $\mathcal{B}_1$ and $\mathcal{B}_2$ or modulo β with respect to the option, with as input $a = c \times G\text{”} + v\text{”}$ where $G\text{”} = 2G' \times M_1' \bmod D_G$ and $v\text{”} = (M_1' \times \det G)v_1 \bmod D_G$. Thus we propose the following procedure for computing the Closest Vector $\lfloor cG^{-1} \rceil \times G$.

Algorithm 4 BabaïROff_rns(option)

```
Input: $c \in \mathbb{Z}^n$ the ciphertext given in $\mathcal{B}_1$, $\mathcal{B}_2$ and $\widehat{m}$, all the
    values concerned by $G$ are considered as precomputed.
Output: $[(2cG' + (\det G)v_1) \bmod (2 \det G)]$, if (option = rns) then in the two RNS bases $\mathcal{B}_1$ and $\mathcal{B}_2$, else modulo
    $\beta$ (that is true for all the calculus of this procedure).
1: $a \leftarrow c \times G\text{”} + v\text{”}$ in $\mathcal{B}_1$, $\mathcal{B}_2$ and $\widehat{m}$,
2: $b \leftarrow \mathrm{NewPreBabaïROff\_rns}(a, \mathcal{B}_1, \mathcal{B}_2, \widehat{m})$,
3: $r \leftarrow (a - b)(2 \det G)^{-1}$ in $\mathcal{B}_1$, $\mathcal{B}_2$ and $\widehat{m}$.
```

IV. Overall complexity of RNS Round-off CVP method

A. About the size of RNS bases $\mathcal{B}_1$ and $\mathcal{B}_2$

The main interest of RNS is that it allows to perform computations on large integers independantly on residues having a size which can be chosen and adapted to practical considerations, e.g. in the case of implementation into some embedded systems. A basic RNS product being composed of independant modular products modulo each element of the RNS base, these ones are chosen having a particuliar form which guarantee efficient modular multiplications, e.g. $2^t - c_i$ with $c_i < 2^{t/2}$ [6], [5]. Hence, the complexities of algorithms are given in terms of numbers of elementary modular products.

In order to simplify analysis of algorithms and because it provides more modularity for practical implementations, RNS digits are considered having the same size in the two bases $\mathcal{B}_1$ and $\mathcal{B}_2$ required for RNS Montgomery reductions.

More precisely, since $a = c \times G\text{”} + v\text{”}$, with G” and v” reduced modulo D_G, and because M_1 must verify $M_1 > |a|_\infty / D_G$, then it sufficies to have $M_1 > \ell \times |c|_\infty + 1$. Now, recalling that $c = m + kH$ and that $\log(\max|H_{i,j}|) \in \mathcal{O}(\log(\det G))$ (e.g. when $H = HNF(G)$ with $\det G$ prime), both sizes of $\mathcal{B}_1$ and $\mathcal{B}_2$ are identical, i.e. $M_1, M_2 > D_G$. So we will consider that $|\mathcal{B}_1| = |\mathcal{B}_2| = k$.

B. Complexity of Round-off step

a) Matrix multiplication approach for first extension: $\mathrm{Extension_1Bis}$, based on CRT, can be seen as a matrix multiplication. Let's denote $q_{i,j}^{(s)}$ the j-th residue of q_i for s-th coefficient of vector a (i.e. $i \in \{1,2\}$, $j \in [1,k]$, $s \in [1,\ell]$), $m_j^{(i)}$ the j-th modulus of base $\mathcal{B}_i$, $M_{i,j} = \frac{M_1}{m_i^{(1)}} \bmod m_j^{(2)}$,and $\xi_j^{(s)} = \left| q_{1,j}^{(s)} M_{1,j}^{-1} \right|_{m_j}$. Then the first extension can be efficiently reduced to the following matrix product:

$$\begin{pmatrix} q_{2,1}^{(1)} & .. & q_{2,k}^{(1)} \\ \vdots & .. & \vdots \\ q_{2,1}^{(\ell)} & .. & q_{2,k}^{(\ell)} \end{pmatrix} = \begin{pmatrix} \xi_1^{(1)} & .. & \xi_k^{(1)} \\ \vdots & .. & \vdots \\ \xi_1^{(\ell)} & .. & \xi_k^{(\ell)} \end{pmatrix} \times \begin{pmatrix} M_{1,1} & .. & M_{1,k} \\ \vdots & .. & \vdots \\ M_{k,1} & .. & M_{k,k} \end{pmatrix}$$

If k is chosen to be dividing ℓ, i.e. $\ell = k \times n$, then the product can be performed by using n optimised (and parallelizable) Strassen's like square matrix multiplications [22]. More precisely, the complexity can be reduced to $\mathcal{O}(nk^{2+\epsilon} = \ell k^{1+\epsilon})$ multiplications ($\epsilon \sim 0.8074$ for Strassen's technique), instead of $\mathcal{O}(\ell k^2)$ when ℓ standard CRT based extensions are done.

Other steps of $\mathrm{Extension_1Bis}$ only require $4k\ell$ (resp. 3ℓ) multiplications in $\mathcal{B}_2$ (resp. $\bmod \widehat{m}$).

b) Second extension: A bottleneck of classical Montgomery reduction approach is that the obtained result is not totally reduced. As seen before, it is still the case in RNS. Hence a comparison between r'_2 and D_G must be performed in order to correctly execute the rounding step of Babaï CVP method, and to recover exaclty the original plaintext. A solution is then to compute the coefficients of r'_2 into the MRS associated to base $\mathcal{B}_2$. This transformation needs $\ell \times \frac{k(k-1)}{2}$ multiplications for the whole vector r'_2. Then, a direct comparison between MRS coefficients of r'_2 and D_G is possible, and a subtraction executed if necessary.

Associated to the use of Horner's rule to compute the exact result into $\mathcal{B}_1$ (resp. $\mathrm{mod}\beta$), the second extension has a cost of $\ell \times \frac{3k(k-1)}{2}$ (resp. $\ell \times \frac{(k+2)(k-1)}{2}$) multiplications.

Comparing to first extension, we see that this MRS based extension is a bottleneck for the RNS approach.

c) Complexity of algorithm 4: Steps 1 and 3 of the Round-off algorithm 4 are just usual independant RNS operations, each of them costing respectively $(2k+1)\ell^2$ and $(2k+1)\ell$ multiplications.

The complexity of NewPreBabaïROff_rns algorithm is determined by the necessity to get back to a positional number system to perform a comparison. Although its complexity of $\mathcal{O}(\ell k^2)$ appears to be dominated by the one of vector-matrix computations in step 1, these products can be performed in a fully parallel way, contrary to MRS extensions which are naturally sequential.

C. Complexity of full RNS Round-off CVP procedure

Because modulus β has been chosen such that $2 \times |m|_\infty < \beta$, the round-off procedure just has to provide the value of $\lfloor cG^{-1} \rceil$ modulo β. Once we get it, it remains to multiply it to $G \bmod \beta$ and to subtract the product to the ciphertext in order to recover the original plaintext m. This ultimate vector-matrix product has then just a cost of ℓ^2 multiplications $\mathrm{mod}\beta$.

Finally, we get an asymptotic complexity of $\mathcal{O}(k\ell^2)$ elementary modular multiplications for a full decryption using the proposed RNS method.

V. Conclusion

One interesting feature of this approach comes from the formulae of the $\mathrm{Extension}_1\mathrm{Bis}$ which can be decomposed in matrix products where some fast algorithms like the Strassen one [22] can be used. The main drawback of the current version is due to the necessity to compute exactly the result of the NewPreBabaïROff_rns. The solution of using MRS is not efficient, it would be more interesting to use a Shenoy-Kumaresan approach where the formulae are similar to the ones of $\mathrm{Extension}_1\mathrm{Bis}$.

References

[1] L. Babaï, "On Lovasz' lattice reduction and the nearest lattice point problem", *Combinatorica*, Springer-Verlag, 1986, 6(1), pp.1-13.

[2] J.C. Bajard *et al.*, "Modular multiplication and base extensions in residue number systems", *Proc. 15th IEEE Symp. on Comp. Arithmetic, ARITH*, 2001, pp.59-65.

[3] J.C. Bajard *et al.*, "Residue systems efficiency for modular products summation: application to elliptic curves cryptography", *Proc. Adv. Signal Process. Algo., Architectures, and Implementations, SPIE 16*, San Diego, 2006.

[4] J.C. Bajard and T. Plantard, "RNS bases and conversions", *Proc. SPIE 14*, Denver, 2004, pp.60-69.

[5] J.C. Bajard *et al*, "Selected RNS Bases for Modular Multiplication", *Proc. IEEE 19th ARITH*, Portland, 2009.

[6] J.C. Bajard *et al*, "Efficient RNS bases for Cryptography", *Proc. of Applied Mathematics and Simulation, IMACS?05*, Paris, 2005.

[7] R. Cheung *et al*, "FPGA Implementation of Pairings Using Residue Number System and Lazy Reduction", *Proc. Cryptographic Hardware and Embedded Syst., CHES*, Nara, Japan, 2011, pp.421-441.

[8] L. Ducas and P.Q. Nguyen, "Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures", *Proc. ASIACRYPT*, 2012, Bejin, China, pp.433-450.

[9] F. Gandino *et al*, "A general approach for improving RNS Montgomery exponentiation using pre-processing", *Proc. IEEE 20th ARITH*, Tbingen, Germany, 2011.

[10] C. Gentry, "Fully homomorphic encryption using ideal lattices", *Proc. 41st Symp. on Theory of Computing, STOC*, New-York, 2009, pp.169-178.

[11] C. Gentry *et al*, "Trapdoors for hard lattices and new cryptographic constructions", *Proc. 40th STOC*, Victoria, Canada, 2008.

[12] O. Goldreich *et al*, "Public-key cryptosystems from lattice reduction problems", *Proc. CRYPTO*, 1997, Santa Barbara, pp.112-131.

[13] N. Guillermin, "A High Speed Coprocessor for Elliptic Curve Scalar Multiplications over $\mathbb{F}_p$", *Proc. CHES*, Santa Barbara, 2010.

[14] J. Hoffstein *et al*, "NTRUSIGN: Digital signatures using the NTRU lattice", *Proc. RSA conf. on The cryptographers' track*, 2003, pp.122-140.

[15] S Kawamura *et al*, "Cox-Rower Architecture for Fast Parallel Montgomery Multiplication", *Proc. EUROCRYPT*, Bruges, Belgium, 2000.

[16] P. Klein, "Finding the closest lattice vector when it's unusually close", *Symp. on Discrete Algorithms, SODA*, San Francisco, 2000.

[17] D. Micciancio, "Improving lattice-based cryptosystems using the Hermite normal form", *Proc. Cryptography and Lattices Conf.*, Providence, 2001.

[18] P. Montgomery, "Modular multiplication without trial division", *Math. of Comp.*, 1985, 44(170), pp.519-521.

[19] K.C. Posch and R. Posch, "Modulo reduction in residue number systems", *IEEE Trans. Parallel Distrib. Syst.*, 1995, 6(5), pp.449-454.

[20] O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography", *J. ACM*, 2009, 56(6).

[21] P.P. Shenoy and R. Kumaresan, "Fast Base Extension Using a Redundant Modulus in RNS", *IEEE Trans. Comput.*, 1989, 38(2), pp.292-297.

[22] V. Strassen, "Strassen. Gaussian elimination is not optimal", *Numerische Mathematik*, 1969, 13, pp.354-356.

[23] N.S. Szabo and R.I. Tanaka, "Residue Arithmetic and its Applications to Compututer Technology", McGraw-Hill, 1967.

[24] A. Svoboda and M. Valach, "Operational Circuits. Stroje na Zpracovani Informaci", Sbornik III, Nakl. CSAV, Prague, 1955, pp.247-295.