Babaï Round-Off CVP method in RNS Application to Lattice based cryptographic protocols

Jean-Claude Bajard, Julien Eynard, Nabil Merkiche, Thomas Plantard

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, SCSSE
University 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).**

I. Introduction

The cryptography based on lattices appeared at the beginning of this century with initial propositions like GGH [10] and NTRU [12].

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 [8]. Despite numerous attacks against one the historical propositions, countermeasure after countermeasure, these systems are still available [6]. 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 [14], [9], [18], 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 Systems (RNS) has been proved to be efficient for other crpyotgraphic systems [11], [5], 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 using a strongly reduced basis given by a matrix G and we construct another basis $H = UG$, where U is a unimodular matrix such that H is a bad basis in terms of lattice basis reduction. H can be in Hermite Normal Form [15].

The encryption mode [10], 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 $c = (c_0, 0, ..., 0)$ with c_0 huge, or $c = (c_0, c_1, ..., c_n)$, c_i smaller. 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 [21], [20]. 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 $M = \prod_{i=1}^{n} m_i$. In this approach we use the modular reduction proposed by P. Montgomery [16] and adapted to RNS [17], [13], [2], both for the evaluation of $\lfloor cG^{-1} \rceil \times G$, and for the final reduction $\bmod \beta$.

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 $\bmod(2 \det G)$, 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 $M_2 = \prod_{m \in \mathcal{B}_2} m$. The bases are selected such that $D_G < M_1, M_2$, 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.

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 $\bmod \beta$.

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: $q_2 \leftarrow q_1$ 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 the base $\mathcal{B}_1$ if *option*=*rns*, else modulo β.

The "PreBabaïROffrns" 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.

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. But it is costly. So we can replace steps 2 and 3 by an approach where we use an extra modulo $\widehat{m}$.

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

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

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

Hence, $r_2 < (2 + \alpha)D_G$, 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 $\widehat{m} > |\mathcal{B}_1| + 1 \geq 2 + \alpha$.

Algorithm 2 Extension$_1$Bis

Input: a_2 defined on $\mathcal{B}_2$ and $a_{\widehat{m}} = a \bmod \widehat{m}$.

Output: q_2 the extension of q_1 in $\mathcal{B}_2$ with $q_2 < M_1$

1: $q_2 \leftarrow \displaystyle\sum_{m \in \mathcal{B}_1} \left| q_{1,i} \left| \frac{M_1}{m_i} \right|_{m_i}^{-1} \right|_{m_i} \frac{M_1}{m_i}$ in $\mathcal{B}_2$

and

$q_{\widehat{m}} \leftarrow \displaystyle\sum_{m \in \mathcal{B}_1} \left| q_{1,i} \left| \frac{M_1}{m_i} \right|_{m_i}^{-1} \right|_{m_i} \frac{M_1}{m_i} \bmod \widehat{m}$

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 $\widehat{m}$ 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 $\mathcal{B}_2$

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

C. Analysis of the second extension

For the second base extension, we can use an extra modulo $\widehat{m}$ with a Shenoy-Kumaresan approach [19]. 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 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 "Rounding 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 which includes the results of the extensions analysis.

NewPreBabaïROff_rns algorithm gives $\lfloor cG^{-1} \rceil$ in the two bases $\mathcal{B}_1$ and $\mathcal{B}_2$ or modulo β with respect to the option, with as input $a = c \times G" + v"$ where $G" = 2G' \times M_1' \bmod D_G$ and

Algorithm 3 NewPreBabaïROff_rns(*option*)

Input: $a = c \times G" + v"$, $a \in \mathbb{Z}^n$ 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: $r'_2 \leftarrow \text{Extension}_1\text{Bis}(q_1, \mathcal{B}_1, \mathcal{B}_2, \widehat{m})$,
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 the base $\mathcal{B}_1$ if *rns* else modulo β,
6: Subtraction of D_G if necessary.

$v" = (M'_1 \times \det G)v_1 \bmod D_G$. 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: $r = \left\lfloor \frac{cG'}{\det G} \right\rceil = \left\lfloor \frac{cG'}{\det G} + \frac{1}{2}v_1 \right\rfloor$, if (option = rns) then in the two RNS bases $\mathcal{B}_1$ and $\mathcal{B}_2$, else modulo β (that is true for all the calculus of this procedure).

1: $a \leftarrow c \times G" + v"$ in $\mathcal{B}_1$, $\mathcal{B}_2$ and $\widehat{m}$,
2: $b \leftarrow \text{NewPreBabaïROff\_rns}(a, \mathcal{B}_1, \mathcal{B}_2, \widehat{m})$,
3: $r \leftarrow (a - b)(2 \det G)^{-1}$ in $\mathcal{B}_1$, $\mathcal{B}_2$ and $\widehat{m}$.

IV. Discussions

One interesting feature of this approach comes from the formulae of the $\text{Extension}_1\text{Bis}$ which can be decomposed in matrix products where some fast algorithms like the Strassen one 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 $\text{Extension}_1\text{Bis}$.

References

[1] L Babai, *On Lovasz' lattice reduction and the nearest lattice point problem.* In Combinatorica, 6(1):1–13, 1986.

[2] JC Bajard, LS Didier and P Kornerup, *Modular multiplication and base extensions in residue number systems* 15th IEEE Symposium on Computer Arithmetic, 2001, pp:59-65

[3] JC Bajard, S Duquesne, M Ercegovac and N Meloni, *Residue systems efficiency for modular products summation: application to elliptic curves cryptography,* SPIE, Adv. Signal Proc. Algo., Archi., and Impl., 2006.

[4] JC Bajard and Th Plantard, *RNS bases and conversions* SPIE, Adv. Signal Proc. Algo., Archi., and Impl., 2004.

[5] R Cheung, S Duquesne, J Fan, N Guillermin, I Verbauwhede and G Yao, *FPGA Implementation of Pairings Using Residue Number System and Lazy Reduction,* CHES 2011, LNCS 6917, 421-441, 2011.

[6] L Ducas and PQ Nguyen, *Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures.* 2012 ASIACRYPT , Bejin, China, pp:433-450.

[7] F Gandino, F Lamberti, P Montuschi, and J C. Bajard, *A general approach for improving RNS Montgomery exponentiation using pre-processing,* Proc. IEEE 20th Symposium on Computer Arithmetic, july 2011.

[8] C Gentry, *Fully homomorphic encryption using ideal lattices,* In Proc. STOC-09, pages 169-178. ACM, 2009.

[9] C Gentry, V Vaikuntanathan and Ch Peikert *Trapdoors for hard lattices and new cryptographic constructions,* STOC 2008, 14th ACM symposium on Theory of computing

[10] O. Goldreich, S. Goldwasser and S. Halevi, *Public-key cryptosystems from lattice reduction problems,* In Proc. of Crypto'97, volume 1294 of LNCS.

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

[12] J. Hoffstein, N. A. H. Graham, J. Pipher, J. H. Silverman, and W. Whyte. *NTRUSIGN: Digital signatures using the NTRU lattice.* CT-RSA, vol 2612 of LNCS, 2003.

[13] S Kawamura, M Koike, F Sano and Ai Shimbo, *Cox-Rower Architecture for Fast Parallel Montgomery Multiplication,* Proc. EUROCRYPT 2000, LNCS 1807.

[14] Ph Klein, *Finding the closest lattice vector when it's unusually close,* SODA '00 Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms.

[15] D Micciancio Improving lattice-based cryptosystems using the Hermite normal form, In Proc. of CALC'01, volume 2146 of LNCS. Springer-Verlag, 2001.

[16] Peter Montgomery, *Modular multiplication without trial division.* Math. Comp. 44:170 (1985)

[17] KC Posch and R Posch, *Modulo reduction in residue number systems,* Parallel and Distributed Systems, IEEE Transactions (Vol. 6).

[18] Oded Regev, *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography,* Journal of the ACM (JACM) JACM, Volume 56 Issue 6, September 2009.

[19] PP Shenoy and R Kumaresan, *Fast Base Extension Using a Redundant Modulus in RNS* Journal IEEE Transactions on Computers Volume 38 Issue 2, 1989.

[20] NS Szabo and RI.Tanaka, *Residue Arithmetic and its Applications to Computer Technology,* McGraw-Hill, 1967

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