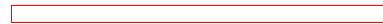


# Modular Multiplication and Base Extensions in Residue Number Systems



Jean-Claude Bajard, Laurent-Stéphane Didier and Peter Kornerup

LIRMM Montpellier , LIP6 Paris , Odense University  
EUROPE

## Introduction

- Modular multiplication is a fundamental operation in cryptography
- Furthermore, for some like RSA we need modular exponentiation
- Cryptographic keys are very huge numbers (around one thousand digits)
- Question: Which is the most efficient representation?

## Residue Number System

Let  $m_1, m_2, \dots, m_{n-1}, m_n$  relatively prime numbers and  $M = \prod_{i=1}^n m_i$ .

- We can represent  $X \in [0, M]$  with  $(x_1, x_2, \dots, x_n)$  such that:

$$\begin{cases} x_1 = X \bmod m_1 \\ x_2 = X \bmod m_2 \\ \vdots \\ x_n = X \bmod m_n \end{cases}$$

- furthermore each  $(x_1, x_2, \dots, x_n)$  such that  $x_i \in [0, m_i - 1]$  represents an unique  $X \in [0, M]$ .  
 $(m_1, m_2, \dots, m_n)$  is named RNS base, we denote it  $\mathcal{B}_n$ .

## Operations

$$\mathbf{X}_{RNS} = (x_1, x_2, \dots, x_{n-1}, x_n)_{RNS}$$

$$\mathbf{Y}_{RNS} = (y_1, y_2, \dots, y_{n-1}, y_n)_{RNS}$$

$$\mathbf{X}_{RNS} + \mathbf{Y}_{RNS} = ((x_1 + y_1) \bmod m_1, \dots, (x_n + y_n) \bmod m_n)_{RNS}$$

$$\mathbf{X}_{RNS} \times \mathbf{Y}_{RNS} = ((x_1 \times y_1) \bmod m_1, \dots, (x_n \times y_n) \bmod m_n)_{RNS}$$

**Advantage :** integral parallel computing **Drawback:** comparisons, overflows

## Exact division

- If  $\gcd(\mathbf{Y}, \mathbf{M})=1$  then

$$\mathbf{Y}_{\mathbf{M}}^{-1} = ((y_1)_{m_1}^{-1}, (y_2)_{m_2}^{-1}, \dots, (y_{n-1})_{m_{n-1}}^{-1}, (y_n)_{m_n}^{-1})$$

is the inverse of  $\mathbf{Y}$  modulo  $\mathbf{M}$ .

Now, if  $\mathbf{Y}$  is invertible and if  $\mathbf{X}$  is a multiple of  $\mathbf{Y}$ , then:  $\frac{\mathbf{X}}{\mathbf{Y}} = \mathbf{X}\mathbf{Y}_{\mathbf{M}}^{-1}$

## Reconstruction with the Chinese Remainder Theorem

- $\mathbf{X}_{RNS} = (x_1, x_2, \dots, x_n)$  in  $\mathcal{B}_n$  with  $X \in [0, M[$ .

we have

$$X = \left( \sum_{i=1}^n x_i |M_i|_{m_i}^{-1} M_i \right) \bmod M$$

with  $M_i = \frac{M}{m_i}$ , and  $|M_i|_{m_i}^{-1}$  inverse of  $M_i$  modulo  $m_i$ .

- Remark:

$$(x_i |M_i|_{m_i}^{-1} M_i) \bmod m_j = x_i \text{ if } j = i$$

$$(x_i |M_i|_{m_i}^{-1} M_i) \bmod m_j = 0 \text{ else}$$

## Mixed Radix

- $X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \dots + a_n m_1 \dots m_{n-1}$
- MRS Representation  $X = (a_1, a_2, a_3, \dots, a_n)$

$$\begin{array}{l}
 \bullet \left\{ \begin{array}{l}
 a_1 = x_1 \bmod m_1 \\
 a_2 = (x_2 - a_1) m_{1,2}^{-1} \bmod m_2 \\
 a_3 = ((x_3 - a_1) m_{1,3}^{-1} - a_2) m_{2,3}^{-1} \bmod m_3 \\
 a_4 = (((x_4 - a_1) m_{1,4}^{-1} - a_2) m_{2,4}^{-1}) - a_3) m_{3,4}^{-1} \bmod m_4 \\
 \vdots \\
 a_n = (\dots (x_n - a_1) m_{1,n}^{-1} - a_2) m_{2,n}^{-1}) - \dots - a_{n-1}) m_{n-1,n}^{-1} \bmod m_n
 \end{array} \right.
 \end{array}$$

where  $m_{i,j}^{-1}$  is the inverse of  $m_i$  modulo  $m_j$

## RNS to RNS base conversion

- With Mixed Radix: for each  $\tilde{m}_i$  we evaluate

$$\tilde{x}_i = |a_1 + a_2 m_1 + a_3 m_1 m_2 + \dots + a_n m_1 \dots m_{n-1}|_{\tilde{m}_i}$$

- Shenoy et Kumaresan :

we have, 
$$\left( \sum_{i=1}^n M_i \left| |M_i|_{m_i}^{-1} x_i \right|_{m_i} \right) = X + \alpha \times M$$

$$\alpha = \left| |M|_{m_{n+1}}^{-1} \left( \sum_{i=1}^n \left| M_i \left| |M_i|_{m_i}^{-1} x_i \right|_{m_i} \right|_{m_{n+1}} - |X|_{m_{n+1}} \right) \right|_{m_{n+1}}$$

$$\tilde{x}_j = \left| \sum_{i=1}^n \left| M_i \left| |M_i|_{m_i}^{-1} x_i \right|_{m_i} \right|_{\tilde{m}_j} - |\alpha M|_{\tilde{m}_j} \right|_{\tilde{m}_j}$$

## Montgomery Algorithm(1985)

### Algorithme 1

*evalprod(A,B,N,S)*

$S \leftarrow 0$

for  $i = 0$  to  $n - 1$  do

$q_i \leftarrow ((s_0 + a_i * b_0)(\beta - n_0)^{-1} \bmod \beta$

$S \leftarrow S + a_i * B + q_i * N$

$S \leftarrow S \div \beta$

$A, B, Q, S$  et  $N$  huge numbers,  $\beta$  is the radix.

And,  $(\beta - n_0)^{-1} * n_0 \equiv -1 \bmod \beta$ .

## Montgomery Algorithm(1985)

**Remark :** the evaluated value is such that  $S (< 2N)$

$$\frac{(A * B + Q * N)}{\beta^n} \text{ in other words } A \times B \times (\beta^n)_N^{-1} \bmod N$$

**Second Pass :** with  $\beta^{2n} \bmod N$  as input, to obtain  $A \times B \bmod N$ .

## Transfer to RNS

1

- We have  $\mathcal{B}_n = (m_1, m_2, \dots, m_{n-1}, m_n)$  as RNS base, and  $M = \prod_{i=1}^n m_i$ .
- We have to find  $Q$ ,  $Q < M$ , such that:  $A * B + Q * N$  is a multiple of  $M$ .
- Thus the RNS representation of  $A * B + Q * N$  in  $\mathcal{B}_n$  is composed only of zeros.
- For  $i = 1..n$ ,  $(a_i * b_i + q_i * n_i) \bmod m_i = 0$ ,
- We can construct  $Q < M$ ,  
for  $i = 1..n$ , we have  $q_i = (m_i - a_i * b_i) * (n_i)_{m_i}^{-1} \bmod m_i$ .

---

<sup>1</sup>first occurred in Posh&Posh 1995

Auxiliary base for  $\mathbf{R} = (\mathbf{A} * \mathbf{B} + \mathbf{Q} * \mathbf{N}) * \mathbf{M}^{-1}$

- $\mathbf{A} * \mathbf{B} + \mathbf{Q} * \mathbf{N}$  is multiple of  $\mathbf{M}$
- thus  $\mathbf{R} = (\mathbf{A} * \mathbf{B} + \mathbf{Q} * \mathbf{N}) * \mathbf{M}^{-1}$  is possible in RNS in a base  $\tilde{\mathcal{B}}_{\tilde{n}}$  where  $\mathbf{M}$  is invertible,
- $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{N}$  must be known in  $\tilde{\mathcal{B}}_{\tilde{n}}$
- and  $\mathbf{Q}$  needs to be represented in this new base.

## Algorithme 2 (RNS Modular Multiplication)

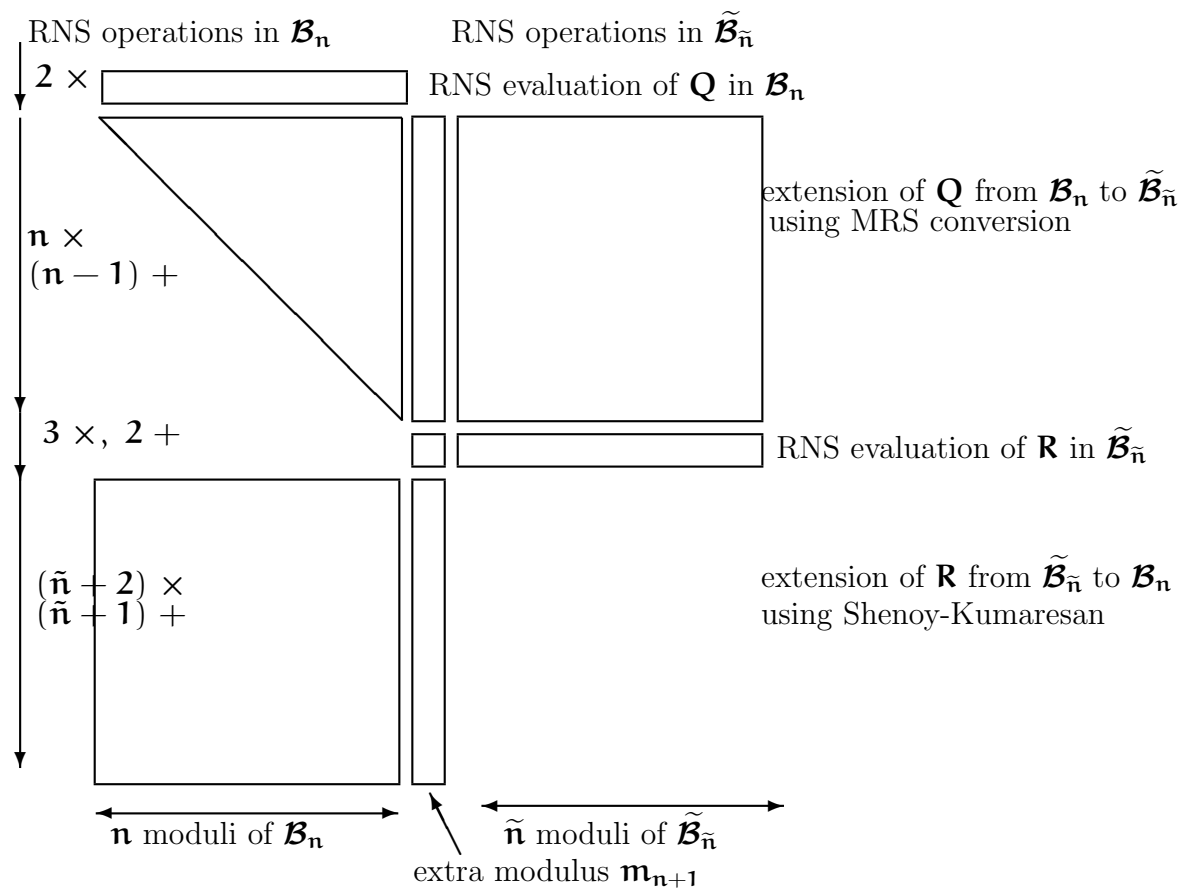
**Function:** *RNS\_Modular\_Montgomery\_Multiplication*

**Stimulus:** *Bases:  $\mathcal{B}_n$ ,  $\{m_1, m_2, \dots, m_n\}$ , and  $\tilde{\mathcal{B}}_{\tilde{n}}$ ,  $\{\tilde{m}_1, \tilde{m}_2, \dots, \tilde{m}_{\tilde{n}}\}$ , where  $M = \prod_{i=1}^n m_i$ ,  $\tilde{M} = \prod_{i=1}^{\tilde{n}} \tilde{m}_i$   $\gcd(M, \tilde{M}) = 1$ ,  $M < \tilde{M}$  integers  $N$ ,  $A$  and  $B$  expressed in RNS in the two bases, with  $\gcd(N, M) = 1$ ,  $\gcd(N, \tilde{M}) = 1$ , and  $0 < 2N < M$  with  $A * B < M * N$*

**Response:** *An integer  $R < 2N$  expressed in the two RNS bases such that  $R \equiv ABM^{-1} \pmod{N}$*

**Method:**  *$Q \leftarrow (-A \times_{\text{RNS}} B) \times_{\text{RNS}} N^{-1}$  in  $\mathcal{B}_n$   
Conversion of the representation of  $Q$  from  $\mathcal{B}_n$  to  $\tilde{\mathcal{B}}_{\tilde{n}}$   
 $R \leftarrow (A \times_{\text{RNS}} B +_{\text{RNS}} Q \times_{\text{RNS}} N) \times_{\text{RNS}} M^{-1}$  in  $\tilde{\mathcal{B}}_{\tilde{n}}$   
Conversion of the representation of  $R$  from  $\tilde{\mathcal{B}}_{\tilde{n}}$  to  $\mathcal{B}_n$*

## Evaluation of $\mathbf{A} \times \mathbf{B} \times \mathbf{M}^{-1}$ in RNS



## Allowing an offset in the residue

By the CRT

$$\widehat{Q} = \sum_{i=1}^n \left| q_i |M_i|_{m_i}^{-1} \right|_{m_i} M_i = Q + \alpha M \quad (1)$$

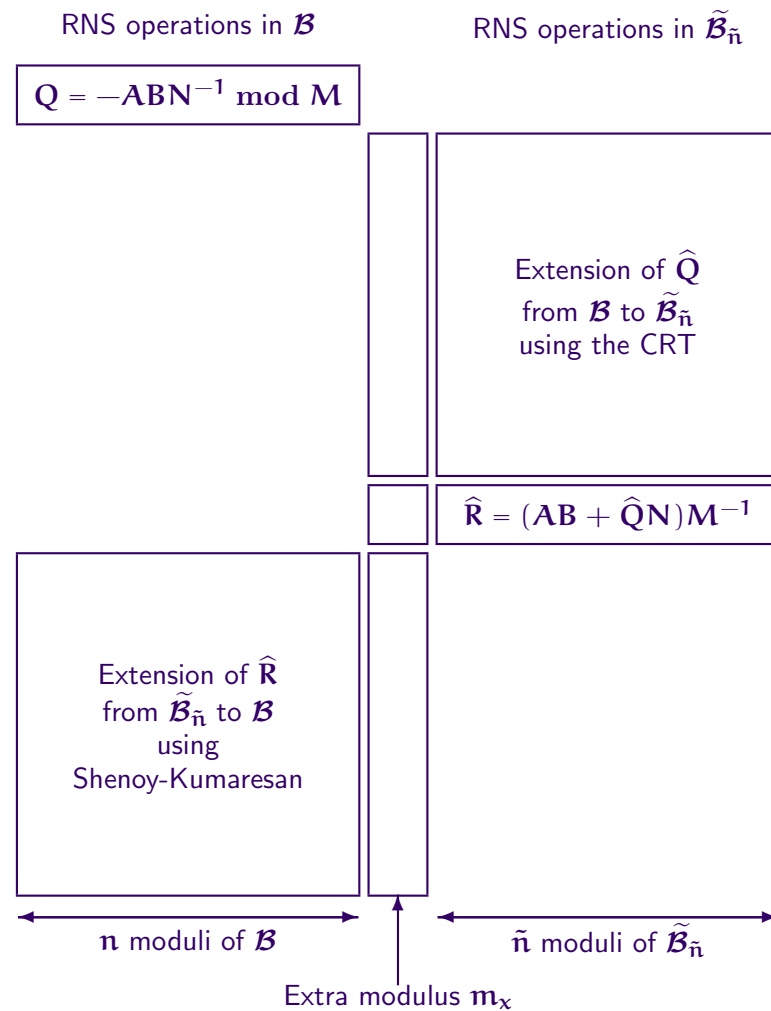
for some value of  $\alpha$  where  $0 \leq \alpha < n$ .

When  $\widehat{Q}$  has been computed it is possible to compute  $\widehat{R}$  as

$$\begin{aligned} \widehat{R} &= (AB + \widehat{Q}N)M^{-1} = (AB + QN + \alpha MN)M^{-1} \\ &= (AB + QN)M^{-1} + \alpha N \end{aligned}$$

so that  $\widehat{R} \equiv R \equiv ABM^{-1}(\text{mod } N)$ , which is sufficient for our purpose.  
Also, assuming that  $AB < NM$  we find that  $\widehat{R} < (n + 2)N$  since  $\alpha < n$ .

# Evaluation of $\mathbf{A} \times \mathbf{B} \times \mathbf{M}^{-1}$ in RNS



## Example

We consider the systems  $\mathcal{B}_5 = \{3, 7, 13, 19, 29\}$ ,  $\tilde{\mathcal{B}}_5 = \{5, 11, 17, 23, 31\}$ , the extra modulus  $m_e = 8$  and operands  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{N}$ . Thus, we have  $\mathbf{M} = 150423$  and  $\tilde{\mathbf{M}} = 666655$ .

	In $\mathcal{B}_n$	$m_x$	In $\tilde{\mathcal{B}}_n$	Base 10
	3 7 13 19 29	8	5 11 17 23 31	
<b>A</b>	1 3 9 14 25	2	1 8 2 5 5	26386
<b>B</b>	1 5 1 9 25	3	1 1 1 21 19	72931
<b>N</b>	1 2 6 11 27	7	2 7 9 14 19	14527

The computation of  $\mathbf{A} \times \mathbf{B} \times \mathbf{M}^{-1} \bmod \mathbf{N}$  is detailed as shown in the following table.

In $\mathcal{B}_n$	$m_x$	In $\tilde{\mathcal{B}}_n$	Computation
2 3 5 11 8			$\mathbf{Q} \leftarrow (-\mathbf{A} \times \mathbf{B}) \times \mathbf{N}^{-1} = 143993$
	5	1 3 8 1 14	Extension of Q from $\mathcal{B}_n$ to $\tilde{\mathcal{B}}_n$
	1	2 5 12 6 7	
	0	2 0 16 16 16	
	6	0 2 14 17 29	
	7	4 10 0 19 20	
			$\hat{\mathbf{Q}} = 444839 = 143993 + 2 * 150423$
	1	3 5 10 1 15	$\hat{\mathbf{R}} \leftarrow (\mathbf{A} \times \mathbf{B} + \hat{\mathbf{Q}} \times \mathbf{N}) \times \mathbf{M}^{-1}$
0 6 9 5 25	1		Conversion of $\mathbf{R}$ through Shenoy and Kumaresan algorithm
2 3 12 12 4	3		
2 2 2 6 17	5		
0 2 10 13 10	3		
1 0 2 13 24	6		
	3		
1 5 9 7 15			$(= \alpha)$ $\hat{\mathbf{R}} = 55753 = (12172 + 3 * \mathbf{N}) \bmod \mathbf{M}$ $= (\mathbf{A} \times \mathbf{B} \times \mathbf{M}^{-1} \bmod \mathbf{N} + 3 * \mathbf{N}) \bmod \mathbf{M}$
1 1 9 2 23	4	0 7 0 22 25	$\mathbf{M}^2 \bmod \mathbf{N} = 12580$ with $\hat{\mathbf{R}}$ as input
			Montgomery with exact extension
2 6 9 11 19			$\mathbf{AB} \bmod \mathbf{N} = 9257$

### Trick for 32 bits units

$m$  prime, and for  $0 \leq a < m < 2^{32}$  let  $[a]_m = a 2^{32} \bmod m$   
 then  $[a + b]_m = |[a]_m + [b]_m|_m$  where the outer reduction is an ordinary  
 reduction modulo  $m$ .

### Algorithme 3 Montgomery's $M$ -reduce( $t$ )

**Stimulus:** An integer  $t$  such that  $0 \leq t < rm$ .

$m, r, m', r^{-1}$  such that  $\gcd(m, r) = 1, r > m > 2$  and  $rr^{-1} - mm' = 1$ .

**Response:** An integer  $u, u = (tr^{-1}) \bmod m$ .

**Method:**  $q := ((t \bmod r)m') \bmod r;$

$u := (t + qm) \operatorname{div} r;$

if  $u \geq m$  then  $u := u - m;$

With  $r = 2^{32}$ ,  $M\text{-reduce}([a]_m[b]_m) = ab 2^{32} \bmod m = [ab]_m$ , the product  
 $[ab]_m$  can be computed with 3 ordinary 32-bit multiplications  
 Mapping into and out performed by  $M\text{-reduce}, |r^2|_m$ , respectively 1.

## Conclusion

- The main advantage is that we only need identical integer channels  
no pseudo floating point unit as in Posh&Posh (1995) or KKSS (2000)
- conditions are easy to satisfy  
no condition of the choice of the moduli (that is the case in KKSS)  
no condition on minimal value for  $\mathbf{N}$  (that is not the case in Posh&Posh)  
thus the same structure can be used for all the values less than a maximal one
- the number of channels is reduced to  $\mathbf{max}(n, \tilde{n})$
- Less parallel steps and less total numbers operations than in Posh&Posh (1995) or KKSS (2000)

### Posh&Posh conditions

$$\mathbf{N} + \mathbf{N}/6 < \mathbf{M} < \mathbf{N} + \mathbf{N}/3 \text{ and } 4\mathbf{N} \leq \widetilde{\mathbf{M}} < (4 + 1/12)\mathbf{N}$$

$$\mathbf{A}, \mathbf{B} < \mathbf{N} + \mathbf{N}/6$$

complexity not clearly exposed but refer to conclusion  $3n + 5$  mod mult (55 add)

### KKSS conditions

$$\Delta = n(\epsilon + \delta)$$

where  $\epsilon = \max(2^r - m_i)/2^r$  and  $\delta = \max(x_i \|M_i\|_{m_i}^{-1} - \text{trunc})/m_i$

$\alpha$  corrector for example  $1/2$  and  $0 \leq \Delta \leq \alpha < 1$

$$4\mathbf{N} \leq (1 - \Delta)\widetilde{\mathbf{M}} \text{ and } 2\mathbf{N} \leq (1 - \alpha)\mathbf{M}$$

$$\mathbf{A}, \mathbf{B} < 2\mathbf{N}$$

complexity  $2n + 9$  mod mult (32 bits trick does not seems possible)

### Our conditions

$$(2 + n)\mathbf{N} < \mathbf{M}, \widetilde{\mathbf{M}} \text{ for mod mul or } (2 + n)^2\mathbf{N} < \mathbf{M}, \widetilde{\mathbf{M}} \text{ for exp}$$

$$\mathbf{A}, \mathbf{B} < \mathbf{NM}$$

complexity  $(2n + 7)$  mod mult (3 mul with trick)