Montgomery Modular Multiplication via Single-Base Residue Number Systems

Zabihollah Ahmadpour
Dept. of Computer Science and Engineering
Shahid Beheshti University
Tehran, Iran
z_ahmadpour@sbu.ac.ir

Ghassem Jaberipur
Dept. of Computer Engineering
Chosun University
Gwangju, Republic of Korea
Jaberipur@chosun.ac.kr

Jeong-A Lee
Dept. of Computer Engineering
Chosun University
Gwangju, Republic of Korea
jalee@chosun.ac.kr

Abstract— **Montgomery modular multiplication (MMM) in residue number systems (RNS) uses a base extension (BE) technique. This is to avoid division, which is hard, slow and costly in RNS. It is somewhat less costly and faster than the reverse conversion, via Chinese remainder theorem (CRT) and reduction factor method. However, it is used one after the other, for each of the equally large bases. In this work, we modify the conventional RNS-MMM algorithm via replacing the two unparalleled BE undertakings with three parallel CRT-like operations with the same complexity, as BE. As for the reduction factors, we use a special case of the Kawamura's algorithm that leads to definitive result. The proposed RNS-MMM method allows for squaring the working dynamic range, or halving the bit-width of the balanced residue channels. Moreover, the common practice of dynamically changing the working moduli set in security and crypto applications is less critical due to doubled size of the pool of available moduli. The proposed circuits are simulated, tested and synthesized via Synopsys Design Compiler on the TSMC 65-nm technology, to show 69% less delay and 28% less area-time-product at the cost of 14% more energy consumption, with respect to the most relevant reference work.**

Keywords— Montgomery modular multiplication, Residue number systems, Modular reduction factors, Base extension.

I. Introduction

Modular multiplication of large ≥ 1024-bit integers is the basic operation in several public-key cryptosystems (e.g., RSA [1], Rabin [2]). It is often realized via the hardware realization of the well-known Montgomery Modular Multiplication (MMM) algorithm [3].

MMM realization via residue number system (RNS) arithmetic leads to speed gain and low power dissipation, due to the parallel processing nature of residue channels. However, there are some critical issues in deciding the characteristic of the working RNS, as are enumerated below.

1) Equal bit-widths r of the working k-moduli RNS is desirable, since it generally leads to the optimizing property of speed-balance among the parallel residue channels.

2) Crypto key-lengths of over 2^{10} bits and the required counter side-channel attack strategies call for hundreds of co-prime moduli to allow sufficient dynamism in the task of frequently changing the working moduli set [4].

3) The base extension (BE) technique is commonly used for avoiding the difficult, slow and costly division in the MMM. However, the requirement of employing two equally large moduli sets doubles the essential number of co-prime moduli, while the working dynamic range (DR) does not increase and equals that of one moduli set.

4) Deciding on the values of k and r is a critical design issue, where smaller bit-width r, for speedup, results in more number of moduli k, which in turn can slow down the BE.

To take utmost advantage of the available pool of equal bit-width moduli, getting around the necessity of utilizing two bases required by the BE technique, saves half of the moduli in favor of squaring the DR. That is how we were motivated to modify the conventional BE-based RNS-MMM algorithm, via replacing the two unparalleled applications of BE to three parallel operations whose complexities are compatible with that of the Chinese remainder theorem (CRT); hence hereafter referred to as CRT-like operations. Therefore, all the available moduli contribute to the enlargement of DR, where its range is actually squared. Otherwise, the bit-width of the balanced residue channels can be halved, for the same DR. On the other hand, the double sized moduli pool can be best used to further decrease the probability of successful side channel attacks. The simulation, test, and implementation results for the proposed algorithm and the most relevant previous work [5] show advantages in speedup, and area-time (AT) product reduction at the cost of more energy consumption.

In the rest of this paper, some background on MMM definition and RNS essentials are given in Section 2. Section 3 contains the proposed modified RNS-MMM and the definitive reduction factor derivation. The proposed architecture is provided for in Section 4, which is evaluated in Section 5, and its figures of merit are compared with the best previous relevant work. We conclude the paper in Section 6.

II. Background

The Modular multiplication $|X \times Y|_N$ represents the main operation in most public-key cryptographic algorithms (e.g., [1], [2]). The direct realization of $|X \times Y|_N$ ($X, Y \in [0, N-1]$) requires huge hardware resources, since the working modulo N is extremely large. Therefore, the well-known Montgomery modular multiplication (MMM) integer function $MMM(X,Y)$, as is described by (1), is employed to obtain the modular product $|X \times Y|_N$ via (2).

$$Z = MMM(X,Y) = |XY\Gamma^{-1}|_N = \left(XY + N\left|\widetilde{N}XY\right|_\Gamma\right)/\Gamma \quad (1)$$

$$|X \times Y|_N = MMM(Z, |\Gamma^2|_N) \quad (2)$$

The Montgomery factor Γ and its multiplicative inverse Γ^{-1}, satisfy $\Gamma > N$ and $|\Gamma\Gamma^{-1}|_N = 1$, respectively, and $\widetilde{N}$ is the multiplicative and additive inverse of N, as $\widetilde{N} = |(-N)^{-1}|_\Gamma$. A verifying proof of (2) is given in Appendix 1.

Jaberipur's research was supported by the Brain Pool program, funded by the Ministry of Science and ICT, through the National Research Foundation of Korea (RS-2023-00263909), and Lee's research is supported by Basic Science Research Program funded by the Ministry of Education through the National Research Foundation of Korea (NRF-2020R1I1A3063857).

To ease the understanding of the RNS realization of the latter (see Section 3.1), we briefly describe the RNS essentials, as follows.

A. RNS Essentials

A k-moduli RNS is a non-positional number system where a number X is represented by a k-tuple residue $(x_1, x_2, \ldots, x_k)$, with respect to k-tuple moduli $(m_1, m_2, \ldots, m_k)$. A residue x_i, denoted as $x_i = |X|_{m_i}$, is obtained by extracting the integer remainder of $\frac{X}{m_i}$, for $1 \le i \le k$. The cardinality of numbers that are uniquely representable by the RNS in hand, is called dynamic range (DR), which is maximized by securing mutual primality between the k moduli, and thus is equal to $M = m_1 \ldots \times m_k$. Addition, subtraction, and multiplication are performed faster through k parallel residue channels, where the corresponding residue operands are smaller than original binary operands. However, division and comparison are considered as difficult (i.e., slow and costly) operations in RNS, such that often they are performed via reverse conversion of RNS operands to binary, performing wide word binary division or comparison, and forward conversion of the quotient and remainder (if needed) to RNS equivalents [6].

The CRT is often used for the aforementioned reverse conversion, as $X = \left|\sum_{j=1}^{k} \left|x_j M_j^{-1}\right|_{m_j} M_j\right|_M$, where $M_j = \frac{M}{m_j}$, M_j^{-1} is the multiplicative inverse of M_j with respect to m_j. A common way for the latter modulo-M reduction, is described by (3). The reduction factor $0 \le \gamma_X < k$ [7] can be obtained via an implementation method that is explained in Section 3.2.

$$X = \hat{X} - \gamma_X M, \hat{X} = \sum_{j=1}^{k} \left|x_j M_j^{-1}\right|_{m_j} M_j,$$
$$\gamma_X = \left\lfloor \frac{\sum_{j=1}^{k} \left|x_j M_j^{-1}\right|_{m_j} M_j}{M} \right\rfloor \tag{3}$$

III. Proposed algorithm

The conventional RNS MMM algorithm [7] relies on two consecutive applications of the BE technique. The BEs correspond to two non-overlapping moduli sets, whose DRs are greater than N. Such hugely high DR requires large number of moduli with reasonable bit-width of the corresponding residue channels in favor of speeding up the MMM. However, the larger the moduli set, the longer the BE delay and the more its cost. Nevertheless, the proposed new RNS modular multiplication algorithm, as is described in the rest of this Section, consumes the moduli of both bases as one base, and performs three CRT-like operations in parallel; hence one or more of the following possibilities can be in order:

1) Speedup via reducing the bit-widths to half, thus doubling the number of moduli, for the same working DR.

2) More dynamism in moduli set selection to reduce the probability of successful side-channel attacks.

3) Keeping the same bit-widths, but reducing the total number of moduli to half, for faster CRT-like operations.

A. New RNS implementation of (1)

Let the employed single RNS base be denoted as $B = \{m_1, m_2, \ldots, m_k\}$ with k main residue channels of equal width r. This will be augmented with a particular channel corresponding to modulo m_0, whose utility will be explained in the sequel.

Recalling (1), the Montgomery factor Γ is set, as usual (e.g., [7], [8]), to the DR $M = m_1 \times m_2 \ldots \times m_k$ (i.e., $\Gamma = M$). Therefore, $M > N$ and the balanced bit-width r must satisfy $k \times r \ge n = \lceil \log_2 N \rceil$.

At the outset, recalling (1), the integer $Z = \left|\frac{XY + N\left|\tilde{N}XY\right|_M}{M}\right|_N$ can be described as in (4), where $\sigma \in \{0,1\}$ (see Appendix 2, for a proof).

$$Z = \hat{Z} - \sigma N, \hat{Z} = \frac{XY}{M} + \frac{N\left|\tilde{N}XY\right|_M}{M} \tag{4}$$

Recalling (3), and similar equations for Y, we decompose $\hat{Z}$ to Z_1 and Z_2, as follows.

$Z_1 = \frac{XY}{M} = \frac{\hat{X}\hat{Y}}{M} - \gamma_X\hat{Y} - \gamma_Y\hat{X} + \gamma_X M \gamma_Y$, and $Z_2 = \frac{N\Omega}{M} = \frac{N\hat{\Omega}}{M} - \gamma_\Omega N$, where $\Omega = \left|\tilde{N}XY\right|_M = \hat{\Omega} - \gamma_\Omega M$, $\hat{\Omega} = \sum_{j=1}^{k} \left|\omega_j M_j^{-1}\right|_{m_j} M_j$, $\omega_i = |\Omega|_{m_i}$.

$\hat{z_\imath}$, can be obtained in terms of $x_i = |X|_{m_i} = \left|\hat{X}\right|_{m_i}$, $y_i = |Y|_{m_i} = \left|\hat{Y}\right|_{m_i}$, and $\omega_i = |\Omega|_{m_i} = \left|\left|\tilde{N}\right|_{m_i} x_i y_i\right|_{m_i}$, as follows.

$\hat{z_\imath} = \left|\hat{Z}\right|_{m_i} = |Z_1 + Z_2|_{m_i} =$

$\left|\frac{\hat{X}\hat{Y}}{M} - \gamma_X\hat{Y} - \gamma_Y\hat{X} + \gamma_X M \gamma_Y + \frac{N\hat{\Omega}}{M} - \gamma_\Omega N\right|_{m_i} = \left|\frac{\hat{X}\hat{Y} + N\hat{\Omega}}{M} - \left(\gamma_X\hat{Y} + \gamma_Y\hat{X} + \gamma_\Omega N\right)\right|_{m_i}.$

Let $F_i = \left|\frac{\hat{X}\hat{Y} + N\hat{\Omega}}{M}\right|_{m_i}$, and $G_i = \left|\gamma_X\hat{Y} + \gamma_Y\hat{X} + \gamma_\Omega N\right|_{m_i}$, which leads to $\hat{z}_i = |F_i - G_i|_{m_i}$.

Recalling the integer nature of $\hat{Z}$, so must be the fraction F_i. The corresponding integer expression is derived below, using six variables defined as $\xi_{x_i} = |x_i M_i^{-1}|_{m_i}$, $X_i' = \hat{X} - \xi_{x_i} M_i = \sum_{\substack{j=1 \\ j \ne i}}^{k} \xi_{x_j} M_j$, and similar definitions for ξ_{y_i}, Y_i', ξ_{ω_i}, and Ω_i'. Moreover, six similar ones are defined with the index j. Also we use the following identities:

$\left|\frac{X_i'}{M}\right|_{m_i} = \left|\sum_{\substack{j=1 \\ j \ne i}}^{k} \xi_{x_j} m_j^{-1}\right|_{m_i}$. Likewise, $\left|\frac{Y_i'}{M}\right|_{m_i} = \left|\sum_{\substack{j=1 \\ j \ne i}}^{k} \xi_{y_j} m_j^{-1}\right|_{m_i}$, and $\left|\frac{\Omega_i'}{M}\right|_{m_i} = \left|\sum_{\substack{j=1 \\ j \ne i}}^{k} \xi_{\omega_j} m_j^{-1}\right|_{m_i}$.

$F_i = \left|\frac{\left(X_i' + \xi_{x_i} M_i\right)\left(Y_i' + \xi_{y_i} M_i\right) + N\left(\Omega_i' + \xi_{\omega_i} M_i\right)}{M}\right|_{m_i} = \left|\frac{X_i' Y_i' + \xi_{x_i} M_i Y_i' + \xi_{y_i} M_i X_i' + N\Omega_i'}{M} + \frac{\xi_{x_i} M_i \xi_{y_i} M_i + N\xi_{\omega_i} M_i}{M}\right|_{m_i}$

Let $F_{i_1} = \left|\frac{X_i'Y_i' + \xi_{x_i}M_iY_i' + \xi_{y_i}M_iX_i' + N\Omega_i'}{M}\right|_{m_i}$, and $F_{i_2} = \left|\frac{\xi_{x_i}M_i\xi_{y_i}M_i + N\xi_{\omega_i}M_i}{M}\right|_{m_i}$.

Appendix 3 provides for a proof that F_{i_2} yields an integer. Therefore, F_{i_1} must also yield an integer. The corresponding integer expression for F_{i_2} is given as in (5), and that of F_{i_1} is derived below. Note that for notational brevity sake the plain Σ denotes $\sum_{\substack{j=1 \\ j\neq i}}^{k}$, unless otherwise specified.

$$F_{i_2} = \left|\begin{array}{l} \left|\xi_{x_i}\xi_{y_i}|M_i|_{m_i}m_i^{-1} + \xi_{\omega_i}|N|_{m_i}m_i^{-1}\right|_{2^{2r}} \\ +x_iy_i\left|M_i^{-1}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor M_i^{-1} + \left\lfloor\frac{N}{m_i}\right\rfloor \widetilde{N}\right)\right|_{m_i} \end{array}\right|_{m_i} \qquad (5)$$

$$F_{i_1} = \left|\frac{\Sigma\left(\xi_{x_j}M_j\right)\Sigma\left(\xi_{y_j}M_j\right) + x_i\Sigma\left(\xi_{y_j}M_j\right) + y_i\Sigma\left(\xi_{x_j}M_j\right) + N\Sigma\left(\xi_{\omega_j}M_j\right)}{M}\right|_{m_i}$$

$$= \left|x_i\Sigma\left(\xi_{y_j}m_j^{-1}\right) + y_i\Sigma\left(\xi_{x_j}m_j^{-1}\right) + N\Sigma\left(\xi_{\omega_j}m_j^{-1}\right)\right|_{m_i}$$

since $\left|\Sigma\left(\xi_{x_j}M_j\right)\right|_{m_i} = 0$, due to $j \neq i$.

Consequently,

$$\widehat{z_i} = \left|\begin{array}{c} x_i\Sigma\left(\xi_{y_j}m_j^{-1}\right) + y_i\Sigma\left(\xi_{x_j}m_j^{-1}\right) + N\Sigma\left(\xi_{\omega_j}m_j^{-1}\right) \\ +\left|\xi_{x_i}\xi_{y_i}|M_i|_{m_i}m_i^{-1} + \xi_{\omega_i}|N|_{m_i}m_i^{-1}\right|_{2^{2r}} \\ +x_iy_i\left|M_i^{-1}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor M_i^{-1} + \left\lfloor\frac{N}{m_i}\right\rfloor \widetilde{N}\right)\right|_{m_i} \\ -\gamma_Y x_i - \gamma_X y_i - \gamma_\Omega N \end{array}\right|_{m_i} \qquad (6)$$

B. Derivation of the reduction factors

Equation set (7) is a reproduction of (3), where $|x_iM_i^{-1}|_{m_i}$ is replaced by ξ_{x_i}, and M_i/M, by $1/m_i$, with similar expressions for Y and Ω.

$$\gamma_X = \left\lfloor\sum_{i=1}^{k}\frac{\xi_{x_i}}{m_i}\right\rfloor, \gamma_Y = \left\lfloor\sum_{i=1}^{k}\frac{\xi_{y_i}}{m_i}\right\rfloor, \gamma_\Omega = \left\lfloor\sum_{i=1}^{k}\frac{\xi_{\Omega_i}}{m_i}\right\rfloor \qquad (7)$$

Following [7], the equations for reduction factors γ_X and γ_Y, can be elaborated on to lead to the definitive expressions in (12), where $m_i = 2^r - \delta_i$, without loss of generality. However, γ_Ω will be handled separately in Section 3.2.1.

Let $\gamma_X^+ = \sum_{i=1}^{k}\frac{\xi_{x_i}}{m_i}$, which given that $\frac{\xi_{x_i}}{m_i} = \frac{\xi_{x_i}}{2^r} + \frac{\xi_{x_i}}{m_i} - \frac{\xi_{x_i}}{2^r} = \frac{\xi_{x_i}}{2^r} + \frac{2^r\xi_{x_i} - m_i\xi_{x_i}}{2^rm_i} = \frac{\xi_{x_i}}{2^r} + \frac{\xi_{x_i}\delta_i}{2^rm_i}$, can be decomposed to an easy to implement part $\gamma_X^e = \frac{\sum_{i=1}^{k}\xi_{x_i}}{2^r}$ and a difficult part $\gamma_X^d = \sum_{i=1}^{k}\frac{\xi_{x_i}\delta_i}{2^rm_i}$, as $\gamma_X^+ = \gamma_X^e + \gamma_X^d$.

In the sequel we show that, via some restrictions and conditions, we can simplify the reduction factor equation as $\gamma_X = \left\lfloor\gamma_X^e + \frac{1}{2}\right\rfloor$; hence no need to obtain γ_X^d. For example, (8) and (9) provide for one pair of sufficient conditions, since they lead to (10), as follows.

$\gamma_X \leq \gamma_X^+ = \gamma_X^e + \gamma_X^d < \gamma_X + \frac{1}{2} \Longrightarrow \gamma_X + \frac{1}{2} - \gamma_X^d \leq \gamma_X^e + \frac{1}{2} < \gamma_X + 1 - \gamma_X^d$, per (8), and $\gamma_X < \gamma_X + \frac{1}{2} - \gamma_X^d \leq \gamma_X^e + \frac{1}{2} < \gamma_X + 1$, per (9).

$$\gamma_X^+ < \gamma_X + \frac{1}{2} \qquad (8)$$

$$\gamma_X^d < \frac{1}{2} \qquad (9)$$

$$\gamma_X < \gamma_X^e + \frac{1}{2} < \gamma_X + 1 \qquad (10)$$

For (8) to hold, the following is obtained from (3).

$X + \gamma_XM = \sum_{i=1}^{k}\xi_{x_i}M_i \Longrightarrow \frac{X}{M} + \gamma_X = \sum_{i=1}^{k}\frac{\xi_{x_i}}{m_i} = \gamma_X^+ < \gamma_X + \frac{1}{2} \Longrightarrow \frac{X}{M} < \frac{1}{2}$, which is satisfied if $M > 4N$, since $X < 2N$, must hold. On the other hand, let δ_{max} denote the maximum δ_i-value, which leads to the following, where $\delta_1 = \delta_{max}$, $\delta_2 = \delta_{max} - 2$, … $\delta_i = \delta_{max} - 2(i-1)$, … $\delta_k = \delta_{max} - 2(k-1)$.

$\gamma_X^d = \sum_{i=1}^{k}\frac{\xi_{x_i}\delta_i}{2^rm_i} < \sum_{i=1}^{k}\frac{\delta_i}{2^r} < \sum_{i=1}^{k}\left(\frac{\delta_{max} - (2i-2)}{2^r}\right) = \frac{k(\delta_{max} - (k-1))}{2^r}$. Therefore, (9) holds if $\frac{k(\delta_{max} - (k-1))}{2^r} < \frac{1}{2}$, which leads to (11).

$$\delta_{max} < \frac{2^{r-1}}{k} + (k-1) \qquad (11)$$

Consequently, (12) yields the reduction factors γ_X and γ_Y, if $M > 4N$, and δ_i values satisfy (11).

$$\gamma_X = \left\lfloor\frac{1}{2} + \frac{\sum_{i=1}^{k}\xi_{x_i}}{2^r}\right\rfloor, \gamma_Y = \left\lfloor\frac{1}{2} + \frac{\sum_{i=1}^{k}\xi_{y_i}}{2^r}\right\rfloor \qquad (12)$$

1) *Derivation of γ_Ω*

As in the X case, $\frac{\Omega}{M} + \gamma_\Omega = \gamma_\Omega^+ = \sum_{i=1}^{k}\frac{\xi_{\omega_i}}{m_i}$. However, unlike the case of X, where $\frac{X}{M} < \frac{1}{2}$, we have $\Omega = \left|\widetilde{N}XY\right|_M < M \Longrightarrow \frac{\Omega}{M} \in [0,1)$, which leads to (13), and furthermore to (14), as follows.

$$\gamma_\Omega \leq \gamma_\Omega^+ = \gamma_\Omega + \frac{\Omega}{M} < \gamma_\Omega + 1 \qquad (13)$$

$\gamma_\Omega \leq \gamma_\Omega^+ = \gamma_\Omega^e + \gamma_\Omega^d < \gamma_\Omega + 1 \Longrightarrow \gamma_\Omega - \gamma_\Omega^d \leq \gamma_\Omega^e < \gamma_\Omega + 1 - \gamma_\Omega^d$, per (13), and $\gamma_\Omega < \gamma_\Omega + \frac{1}{2} - \gamma_\Omega^d \leq \gamma_\Omega^e + \frac{1}{2} < \gamma_\Omega + 2$, per applying (9), for γ_Ω^d. Therefore,

$$\gamma_\Omega = \left\lfloor\gamma_\Omega^e + \frac{1}{2}\right\rfloor - \alpha,$$
$$\alpha = \begin{cases} 0 & if\ \gamma_\Omega^e + \frac{1}{2} < \gamma_\Omega + 1 \\ 1 & if\ \gamma_\Omega^e + \frac{1}{2} \geq \gamma_\Omega + 1 \end{cases} \qquad (14)$$

Replacing in (6) for the three reduction factors, leads to $\widehat{z_i} = \left|\widehat{z_i'} + |\alpha N|_{m_i}\right|_{m_i}$, where the definitive residue $\widehat{z_i'} = \left|\widehat{z_i} - |\alpha N|_{m_i}\right|_{m_i} =$

$$\left|\begin{array}{c} x_i\Sigma\left(\xi_{y_j}m_j^{-1}\right)+y_i\Sigma\left(\xi_{x_j}m_j^{-1}\right)+N\Sigma\left(\xi_{\omega_j}m_j^{-1}\right) \\ +\left|\xi_{x_i}\xi_{y_i}|M_i|_{m_i}m_i^{-1}+|N|_{m_i}\xi_{\omega_i}m_i^{-1}\right|_{2^{2r}} \\ +x_iy_i\left|M_i^{-1}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor M_i^{-1}+\left\lfloor\frac{N}{m_i}\right\rfloor\widetilde{N}\right)\right|_{m_i} \\ -\left\lfloor\gamma_Y^e+\frac{1}{2}\right\rfloor x_i-\left\lfloor\gamma_X^e+\frac{1}{2}\right\rfloor y_i-\left\lfloor\gamma_\Omega^e+\frac{1}{2}\right\rfloor N \end{array}\right|_{m_i}$$

On the other hand, recalling (4), $\hat{Z}=Z+\sigma N \Longrightarrow \hat{z_i}=\left|z_i+|\sigma N|_{m_i}\right|_{m_i}$ where $0\leq\ Z=\left|\hat{Z}\right|_N<N$. Consequently,

$$\widehat{z}'_i=\left|\left|z_i+|\sigma N|_{m_i}\right|_{m_i}-|\alpha N|_{m_i}\right|_{m_i}=$$

$$\left||Z|_{m_i}+|\sigma N|_{m_i}-|\alpha N|_{m_i}\right|_{m_i}\Longrightarrow$$

$$\widehat{z}'_i=|Z+\sigma N-\alpha N|_{m_i}\Longrightarrow\widehat{Z}'=Z+(\sigma-\alpha)N\Longrightarrow$$

$-N\leq\widehat{Z}'<2N$, since $\sigma,\alpha\in\{0,1\}\Longrightarrow\sigma-\alpha\in\{-1,0,1\}$. To fix the undesired negative interval for $\widehat{Z}'$, let $\widehat{Z}'^{+}=\widehat{Z}'+N\Longrightarrow 0\leq\widehat{Z}'^{+}<3N$. Since this result is used as the input for the next MMM, we need to allow the input range to be as $0\leq X,Y<3N$. Then, $M>6N$ satisfies the conditions $\frac{X}{M}<\frac{1}{2}$ and $\frac{Y}{M}<\frac{1}{2}$ that was required for (12). However, to extend the dynamic range to $6N$, we add a constant modulo $m_0=8$, as the sole power-of-two modulo, which increases the DR to 8 times the original $\prod_{i=1}^{k}(2^r-\delta_i)$, where $\delta_i>0$.

Note that enforcing $\widehat{Z}'^{+}$ in the subsequent MMM operations leads to correct results since, recalling (1), $Z=MMM(X,Y)=|XYM^{-1}|_N\Longrightarrow$
$MMM\left(\widehat{Z}'^{+},|M^2|_N\right)=\left|\widehat{Z}'^{+}|M^2|_N|M^{-1}|_N\right|_N=$
$\left|\widehat{Z}'^{+}M^2M^{-1}\right|_N=\left|\widehat{Z}'^{+}M\right|_N=|(Z+(\sigma-\alpha+1)N)M|_N$
$=|ZM|_N=||XYM^{-1}|_NM|_N=|XYM^{-1}M|_N=|X\times Y|_N.$

To obtain $\widehat{Z}'^{+}$, we use (15) to find its residues $\widehat{z}'^{+}_i=\left|\widehat{Z}'^{+}\right|_{m_i}=\left|\widehat{z}'_i+|N|_{m_i}\right|_{m_i}$.

$$\widehat{z}'^{+}_i=\left|\widehat{z}'_i+N\right|_{m_i}=$$
$$\left|\begin{array}{l} x_i\Sigma\left(\xi_{y_j}m_j^{-1}\right)+y_i\Sigma\left(\xi_{x_j}m_j^{-1}\right)+N\Sigma\left(\xi_{\omega_j}m_j^{-1}\right) \\ +\left|\xi_{x_i}\xi_{y_i}|M_i|_{m_i}m_i^{-1}+|N|_{m_i}\xi_{\omega_i}m_i^{-1}\right|_{2^{2r}} \\ +x_iy_i\left|M_i^{-1}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor M_i^{-1}+\left\lfloor\frac{N}{m_i}\right\rfloor\widetilde{N}\right)\right|_{m_i} \\ -\left\lfloor\gamma_Y^e+\frac{1}{2}\right\rfloor x_i-\left\lfloor\gamma_X^e+\frac{1}{2}\right\rfloor y_i-\left\lfloor\gamma_\Omega^e+\frac{1}{2}\right\rfloor N+N \end{array}\right|_{m_i} \quad (15)$$

Algorithm 1 describes the steps of implementation of (15), whose circuit realization is discussed in Section 4. Note that the subtrahends in Step 4 represent the reduction factors in (15).

Algorithm 1 (New RNS-MMM)

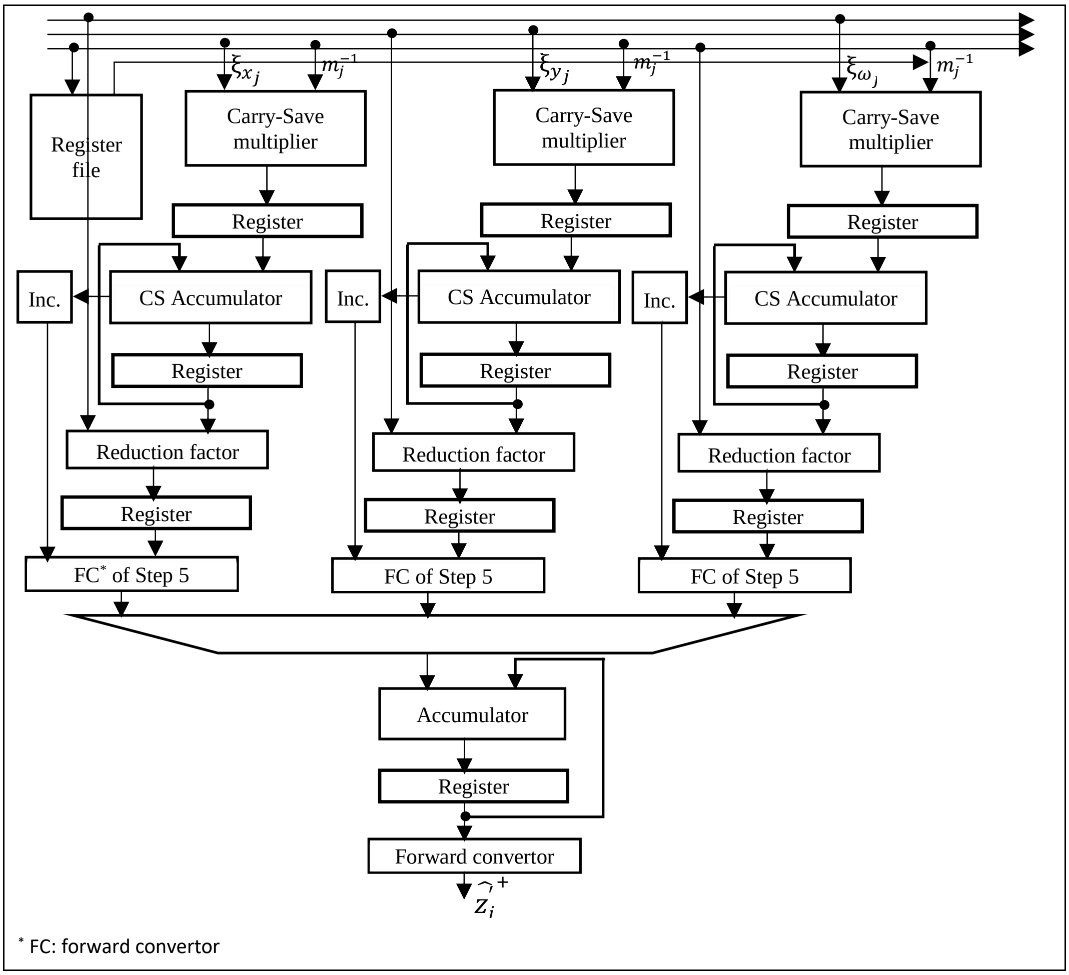
```
Inputs: $m_i$, $x_i$, $y_i$, $0 \leq i \leq k$
Outputs: $\widehat{z}'^{+}_i$, $0 \leq i \leq k$
1) For $i = 0$ to $k$ do par
       $w_i = |x_i y_i|_{m_i}$; $\xi_{x_i} = |x_i M_i^{-1}|_{m_i}$; $\xi_{y_i} = |y_i M_i^{-1}|_{m_i}$;
2) For $i = 0$ to $k$ do par
       $\xi_{\omega_i} = \left|w_i|\widehat{N}M_i^{-1}|_{m_i}\right|_{m_i}$;
       $u_i = \left|w_i\left|M_i^{-1}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor M_i^{-1}+\left\lfloor\frac{N}{m_i}\right\rfloor\widetilde{N}\right)\right|_{m_i}\right|_{m_i}$;
3) For $i = 0$ to $k$ do par
       $s_i = \left|\left|\xi_{x_i}\xi_{y_i}|M_i|_{m_i}m_i^{-1}+\xi_{\omega_i}|N|_{m_i}m_i^{-1}\right|_{2^{2r}}\right|_{m_i}$;
4) For $i = 0$ to $k$ do par
       $p_{x_i} = \left(\sum_{j=0,j\neq i}^{k}\xi_{x_j}m_j^{-1}\right)-\left\lfloor\gamma_X^e+\frac{1}{2}\right\rfloor$;
       $p_{y_i} = \left(\sum_{j=0,j\neq i}^{k}\xi_{y_j}m_j^{-1}\right)-\left\lfloor\gamma_Y^e+\frac{1}{2}\right\rfloor$;
       $p_{\omega_i} = \left(\sum_{j=0,j\neq i}^{k}\xi_{\omega_j}m_j^{-1}\right)-\left\lfloor\gamma_\Omega^e+\frac{1}{2}\right\rfloor$;
5) For $i = 0$ to $k$ do par
       $\widehat{z}'^{+}_i = \left|\begin{array}{c} N+x_i|p_{y_i}|_{m_i}+y_i|p_{x_i}|_{m_i}+ \\ |N|_{m_i}|p_{\omega_i}|_{m_i}+u_i+s_i \end{array}\right|_{m_i}$;■
```

IV. Implementation of Algorithm 1

The main and most complex step of Algorithm 1 is the Step 4, which is mainly implemented by the architecture depicted by Fig. 1. Each of the three parallel expressions for p_{x_i}, p_{y_i}, and p_{ω_i} consist of a main multi-operand MAC (MOMAC) and a rounded reduction factor, which is obtained in parallel (not shown in Fig. 1). Intermediate registers are utilized as well to store the output of the carry-save multiplication and minimize clock cycle delays. Each Σ, in the Step 4 is obtained via the lazy reduction technique [9]. The multipliers are non-modular and produce $2r$-bit products, where the final multiply-accumulate (MAC) results undergo a forward conversion, for deriving $\left|p_{y_i}\right|_{m_i}$, $\left|p_{x_i}\right|_{m_i}$, and $\left|p_{\omega_i}\right|_{m_i}$. The generated 2^{2r}-weighted carries are accumulated to be δ_i^2-folded before entering the forward convertor, since $|2^{2r}c|_{2^r-\delta_i}=|\delta_i^2c|_{2^r-\delta_i}$.

Derivation of the aforementioned Σs, in the channel i, should be preceded with attaining ξ_{x_j}, ξ_{y_j}, and ξ_{ω_j} values, in other channels. To do this, we use the second level MMM, twice, with the Montgomery factor $\Gamma=2^r$ [10], as in (16), which leads to the correct ξ_{x_j}, similarly for ξ_{y_j}, and ξ_{ω_j}.

$$\xi_{x_j}=\frac{\left(x_j\left|M_j^{-1}\right|_{m_j}\right)+m_j\left|(-m_i)^{-1}\left(x_j\left|M_j^{-1}\right|_{m_j}\right)\right|_{2^r}}{2^r} \quad (16)$$



* FC: forward convertor

Fig 1. Three parallel MOMAC implementing (15)

V. Comparison and Evaluation

As it is common in the analytical evaluation of MMM and in crypto algorithms, in general, we discuss the performance of the proposed MMM in terms of number of the required multiplications. However, the overall architecture is synthesized via the Synopsys Design Compiler, and the results compiled in Table 2.

A. *Analytical evaluation*

Table 1 contains the number of required binary multiplications for the Steps of Algorithm 1. The superiority of this work over the best previous one due to [5], is evident via comparing the bottom two rows of Table 1. Equal cost and delay for multipliers of both works are assumed. The reported pipelined cycle counts of [5] is $2k + 22$, which is twice more vs the proposed design. However, for a more fair comparison, the corresponding figure in Table 1 regards the un-pipelined design, which is 6X. Nevertheless, the cost of our work is 50% more, leading to expected 3X cost-speed product.

B. *Synthesis results*

The VHDL codes for designs of the previous most efficient RNS-MMM [5], and the proposed method are mapped into the Synopsys Design Compiler on the TSMC 65 nm standard CMOS library. This was done by enforcing frequency constraints during synthesis.

The results are compiled in Table 2, where some properties and advantages of the proposed work versus that of [5] follows.

Table 1 - # of multiplications required for implementation of (15)

Step	**# of multiplications**	
	CDP⁺	**Total**
1	3	$3\times\ 3 = 9$
2	3	$3 + 3 = 6$
3	2	$3 + 1 = 4$
4	k	$3 \times 3k$
5	3	5
Grand Total	$k + 11$	$k(9k + 24)$
Ref. [5]	$6k + 15$	$k(6k + 15)$

⁺ Critical delay path

- The MMM speed grows, as the channel width increases, since the number of channels decreases for the same key-length.
- 62%, 66%, and 69% delay reductions, for channel widths $r = 24$, 32, and 64, respectively, versus 32 of [5].
- 6% less energy consumption for channel width $r = 24$.
- 39% less area consumption per channel width $r = 24$, and 18% less total area consumption (i.e., including the reduction factors).
- 69%, 42%, and 28% reduced area-time (AT) product, for $r = 24$, $r = 32$, and $r = 64$, respectively.
- Significant reduction in the probability of successful side channel attacks due to increase in the number of available moduli.

TABLE 2 - SYNTHESIS RESULT OF THE SINGLE BASE MODULAR MULTIPLICATION IN COMPARISON WITH THE AUTHOR'S PREVIOUS WORK AND THE MOST SIGNIFICANT PREVIOUS WORKS

Design	**Size of the moduli pool**	**Per residue channel**		r	k	**# of clock cycle**	**Clock cycle time**	**Probability Of successful attacks**	n	MMM **Delay (ns)**	**AT** $(ms \times mm^2)$	**PDP** $(ms \times mW)$
		Area (mm^2)	***Power*** (mW)									
Single Base 3 parallel MOMAC	1981	0.048	31.8	24	43	75	0.76	2^{-294}	1024	57	0.118	0.078
					86	118		2^{-506}	2048	90	0.374	0.246
	384000	0.135	62.3	32	32	64	0.79	2^{-369}	1024	51	0.221	0.102
					64	96		2^{-678}	2048	76	0.660	0.303
	$> 10^6$	0.364	127.9	64	16	48	0.97	$> 2^{-274}$	1024	47	0.274	0.096
					32	64		$> 2^{-520}$	2048	62	0.723	0.253
[5]	251	0.079	17.4	32	32	80	1.86	2^{-134}	1024	149	0.380	0.083

Note that [5] provides for only one channel bit-width $r = 32$, but with two different moduli selection. The smaller moduli set regards moduli of the form $2^{32} - (2^h \pm 1)$, which leads to 251 co-prime moduli. However, the moduli of the form $2^{32} - \delta$, with less restriction on δ, as $\delta < 2^{15}$, provides for 4782 co-prime moduli. Nevertheless, this one is not included in the comparison set, since the corresponding residue channels perform slower than those of the smaller moduli set.

On the other hand, recalling (11), the proposed MMM imposes less restriction on moduli selection, besides co-primality. For example for $r = 32$, $\delta \leq \frac{2^{31}}{32} + 31 = 2^{26} + 31$, leading to 384000 co-prime moduli, out of which $\binom{384000}{32} \approx 10^{144}$ moduli sets of size 32 are dynamically selected. Moreover, the performance of all the corresponding residue channels are based on deferred end-around carry scheme of [11].

VI. CONCLUSION

The conventional Montgomery modular multiplication in residue number system relies on two CRT-like operations with two non-overlapping moduli sets. The second operation depends on the result of the first one thus no time overlap is possible in the course of their executions.

The proposed RNS-MMM implementation uses three parallel CRT-like operations, all on the same moduli set; hence it offers faster Montgomery product generation. Also it frees the second base in favor of doubling the size of moduli pool. This, in turn, increases the number of choices for dynamic switching between the working moduli set; hence reducing the probability of successful side channel attacks in the cryptosystem that perform modular exponentiation based on MMM.

The proposed RNS-MMM algorithm enjoys the definitive derivation of the reduction factors, which is achieved via extending the value of Montgomery factor (i.e., $\Gamma = M$) to over six times the value of Montgomery modulo N (i.e., $\Gamma = M > 6N$). This is actually undertaken via augmenting the moduli-set $\{m_1 \dots, m_k\}$ with $m_0 = 8$.

The best results regarding the proposed MMM scheme occurs for the channel width $r = 24$, and key length $n = 1024$ with over 60% speedup, nearly 20% less area cost, and 6% lower energy consumption, in comparison to the best previous relevant work due to [5].

VII. APPENDICES

Appendix 1 (Proof for double application of MMM)

Rewriting (2), as $|X \times Y|_N = MMM(MMM(X, Y), |\Gamma^2|_N)$, and twice application of $MMM(X, Y) = |XY\Gamma^{-1}|_N$, as follows, provides for the desired verification.

$$MMM(MMM(X, Y), |\Gamma^2|_N) =$$

$$MMM(|XY(\Gamma^{-1})|_N, |\Gamma^2|_N) =$$

$$||XY(\Gamma^{-1})|_N \times |\Gamma^2|_N \times (\Gamma^{-1})|_N =$$

$$||XY(\Gamma^{-1})|_N \times |\Gamma|_N|_N = Z$$

Appendix 2 (Proof of σ ∈ {0,1} in (4))

Recalling (1) and (4), proof of $\sigma \in \{0,1\}$ in $Z = \left|\frac{XY + N|\tilde{N}XY|_M}{M}\right|_N = \frac{XY + N|\tilde{N}XY|_M}{M} - \sigma N$ is in order, where 1), 2) and 3) are used as needed.

1) $|XY|_{\Gamma N} = XY$, since $XY < N^2, \Gamma = M > N \Longrightarrow \Gamma N > N^2 \Longrightarrow XY < MN$.

2) $MM^{-1} - N\tilde{N} = 1 \Longrightarrow M^{-1} = \frac{1 + N\tilde{N}}{M}$.

3) $XY + N\left|XY\tilde{N}\right|_M < N^2 + MN =$

$(N/M + 1)MN < 2MN \Longrightarrow$

$$\left|XY + N\left|XY\tilde{N}\right|_M\right|_{\Gamma N} = XY + N\left|XY\tilde{N}\right|_M - \sigma MN,$$

$\sigma \in \{0,1\}$.

$$Z = |XYM^{-1}|_N = \left|\frac{XY(1 + N\tilde{N})}{M}\right|_N = \frac{M\left|\frac{XY(1+N\tilde{N})}{M}\right|_N}{M} =$$

$$\frac{\left|M\frac{XY(1+N\tilde{N})}{M}\right|_{MN}}{\Gamma} = \frac{|XY(1+N\tilde{N})|_{MN}}{M} = \frac{||XY|_{MN} + |XYN\tilde{N}|_{MN}|_{MN}}{M} =$$

$$\frac{|XY + N|XY\tilde{N}|_M|_{MN}}{M} = \frac{\mathrm{XY} + N|\mathrm{XY}\tilde{N}|_{\mathrm{M}} - \sigma\mathrm{M}N}{M} =$$

$$\frac{XY}{M} + \frac{N|XY\tilde{N}|_M}{M} - \sigma N.$$

Appendix 3 ($F_{i_2} = \left|\frac{(\xi_{x_i} M_i \xi_{y_i} + N\xi_{\omega_i})M_i}{M}\right|_{m_i}$ yields an integer**):**

Proof: Recalling the integer nature of $F_i = \left|\frac{\widehat{X}\widehat{Y} + N\widehat{\Omega}}{M}\right|_{m_i}$, the nominator $F_i^n = \widehat{X}\widehat{Y} + N\widehat{\Omega}$ is a multiple of M, and thus a multiple of m_i leading to $|F_i^n|_{m_i} = 0 \Longrightarrow$

$$|F_i^n|_{m_i} = \left|\begin{matrix}\left(\sum_{j=1}^{k} \xi_{x_j} M_j\right) \times \left(\sum_{j=1}^{k} \xi_{y_j} M_j\right) + \\ N\left(\sum_{j=1}^{k} \xi_{\omega_j} M_j\right)\end{matrix}\right|_{m_i} =$$

$$\left|(\xi_{x_i} M_i)(\xi_{y_i} M_i) + N(\xi_{\omega_i} M_i)\right|_{m_i} =$$

$\left|(\xi_{x_i} M_i \xi_{y_i} + N\xi_{\omega_i})M_i\right|_{m_i} = 0$, since $\left|M_j\right|_{m_i} = 0$, for $j \neq i$. Therefore, $\xi_{x_i} M_i \xi_{y_i} + N\xi_{\omega_i} = \Xi m_i$ is a multiple of m_i. On the other hand, $F_{i_2} = \left|\frac{(\xi_{x_i} M_i \xi_{y_i} + N\xi_{\omega_i})M_i}{M}\right|_{m_i} = \left|\frac{\Xi m_i}{m_i}\right|_{m_i} = |\Xi|_{m_i}$ completes the proof. However, to derive an integer expression for Ξ, we proceed as follows, where r, as before, denotes the width of residue channels.

$$F_{i_2} = |\Xi|_{m_i} = \left|\frac{\xi_{x_i} \xi_{y_i} M_i + N\xi_{\omega_i}}{m_i}\right|_{m_i} =$$

$$\left|\frac{\xi_{x_i} \xi_{y_i}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor m_i + |M_i|_{m_i}\right) + \xi_{\omega_i}\left(\left\lfloor\frac{N}{m_i}\right\rfloor m_i + |N|_{m_i}\right)}{m_i}\right|_{m_i} =$$

$$\left|\left\lfloor\frac{M_i}{m_i}\right\rfloor x_i M_i^{-1} y_i M_i^{-1} + \left\lfloor\frac{N}{m_i}\right\rfloor \widetilde{N} x_i y_i M_i^{-1} + \frac{\xi_{x_i} \xi_{y_i} |M_i|_{m_i} + |N|_{m_i} \xi_{\omega_i}}{m_i}\right|_{m_i} =$$

$$\left|\begin{matrix} x_i y_i \left|M_i^{-1}\left(\left\lfloor\frac{M_i}{m_i}\right\rfloor M_i^{-1} + \left\lfloor\frac{N}{m_i}\right\rfloor \widetilde{N}\right)\right|_{m_i} + \\ \left|\xi_{x_i} \xi_{y_i} |M_i|_{m_i} m_i^{-1} + \xi_{\omega_i} |N|_{m_i} m_i^{-1}\right|_{2^{2r}}\end{matrix}\right|_{m_i},$$

since $\frac{\xi_{x_i} \xi_{y_i} |M_i|_{m_i} + |N|_{m_i} \xi_{\omega_i}}{m_i} < m_i^2 + m_i = (2^r - \delta_i)^2 + (2^r - \delta_i) = 2^{2r} + 2^r + \delta_i^2 - (2^{r+1} + 1)\delta_i = 2^{2r} + (1 - 2\delta_i)2^r + (\delta_i^2 - \delta_i) \leq 2^{2r}$.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.

[2] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Massachusetts Inst of Tech Cambridge Lab for Computer Science1979.

[3] P. L. Montgomery, "Modular multiplication without trial division," *Mathematics of computation*, vol. 44, pp. 519-521, 1985.

[4] J.-C. Bajard, L. Imbert, P.-Y. Liardet, and Y. Teglia, "Leak resistant arithmetic," in *CHES*, 2004, pp. 62-75.

[5] F. Gandino, F. Lamberti, G. Paravati, J.-C. Bajard, and P. Montuschi, "An algorithmic and architectural study on Montgomery exponentiation in RNS," *IEEE Transactions on Computers*, vol. 61, pp. 1071-1083, 2012.

[6] A. R. Omondi and A. B. Premkumar, *Residue number systems: theory and implementation* vol. 2: World Scientific, 2007.

[7] S. Kawamura, M. Koike, F. Sano, and A. Shimbo, "Cox-rower architecture for fast parallel montgomery multiplication," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 2000, pp. 523-538.

[8] J.-C. Bajard and L. Imbert, "A full RNS implementation of RSA," *IEEE Transactions on Computers*, vol. 53, pp. 769-774, 2004.

[9] M. Scott, "Implementing cryptographic pairings," *Lecture Notes in Computer Science*, vol. 4575, p. 177, 2007.

[10] J.-C. Bajard and N. Merkiche, "Double level Montgomery Cox-Rower architecture, new bounds," in *International Conference on Smart Card Research and Advanced Applications*, 2014, pp. 139-153.

[11] Z. Ahmadpour and G. Jaberipur, "Up to 8k-bit Modular Montgomery Multiplication in Residue Number Systems With Fast 16-bit Residue Channels," *IEEE Transactions on Computers*, vol. 71, pp. 1399-1410, 2021.