Montgomery Modular Multiplication via Single-Base Residue Number Systems

Zabihollah Ahmadpour
Ghassem Jaberipur
Jeong-A Lee

31st IEEE International Symposium on Computer Arithmetic

ARITH 2024

Málaga, Spain, June 10-12, 2024.

Outline

- RNS-MMM implementation problems
- Solutions
- RNS-MMM: Conventional Base extension algorithm
- New single-base algorithm coupled with free-δ RNS
- Definitive derivation of the reduction factors
- Advantages of SB-MMM vs. BE-MMM
- Conclusion

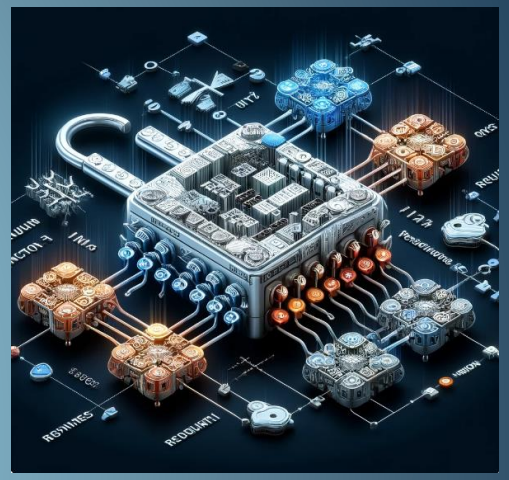
RNS-MMM for Crypto algorithms (e.g., RSA)

- Typical Key-length:

$$n = k \times r = 2^i (9 \le i \le 13)$$

k = # of co-prime moduli,

r = Residue channel bit-width



Source: AI generated image (DALLE)

- Ultra-fast long key-length RNS-MMM via free-δ scheme:

$$r = 16, k \in \{32,64,128,256,512\}$$

- # of dynamically changeable k-moduli pair sets for $k \in \{32,64,128,256,512\}$: $\{2^{505}, 2^{887}, 2^{1302}, 2^{2727}, 2^{3833}\}$

MMM and RNS-MMM

Assumptions:

$\Gamma > N,$

$\mathrm{CGD}(\Gamma, N) = 1,$

$\Gamma_i = |\Gamma|_{m_i},$

$\left|\Gamma_i \times \Gamma_i^{-1}\right|_N = 1,$

$\widetilde{N} = |(-N)^{-1}|_\Gamma$

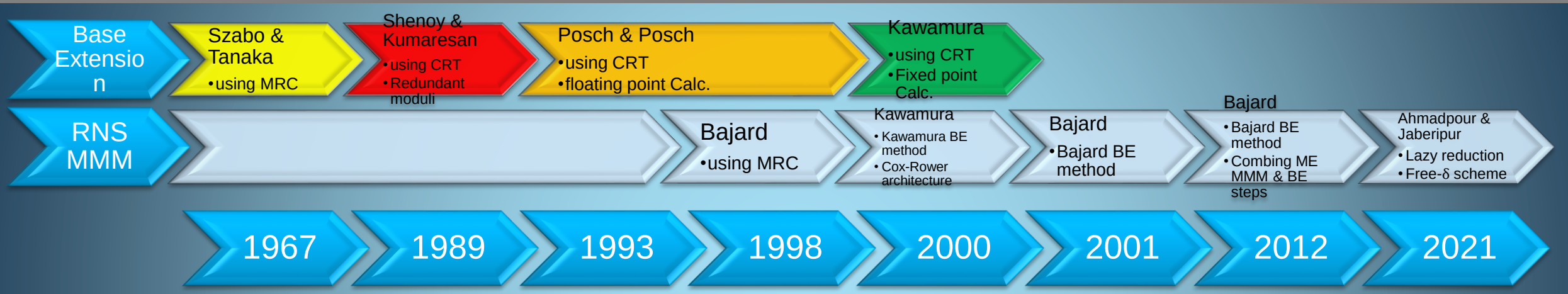
$\widetilde{N}_i = \left|\widetilde{N}\right|_{m_i}$

Algorithm

MMM	RNS-MMM	Problem	Solution
$W = X \times Y$	$w_i = \lvert x_i \times y_i \rvert_{m_i}$		
$Q = \left\lvert W \times \widetilde{N} \right\rvert_\Gamma$	$q_i = \left\lvert \left\lvert W \times \widetilde{N} \right\rvert_\Gamma \right\rvert_{m_i}$	$\left\lvert W \times \widetilde{N} \right\rvert_\Gamma$	$\Gamma = M \Longrightarrow$ $q_i = \left\lvert w_i \times \widetilde{N}_i \right\rvert_{m_i}$
$U = Q \times N$	$u_i = \left\lvert q_i \lvert N \rvert_{m_i} \right\rvert_{m_i}$		
$V = W + U$	$v_i = \lvert w_i + u_i \rvert_{m_i}$		
$Z = V/\Gamma$	$z_i = \lvert v_i \Gamma_i^{-1} \rvert_{m_i}$	$\lvert \Gamma_i^{-1} \rvert_{m_i} = \lvert M^{-1} \rvert_{m_i}^*$	Base Extension

$*$ $\lvert M^{-1} \rvert_{m_i}$ does not exist

Half a century BE development and usage



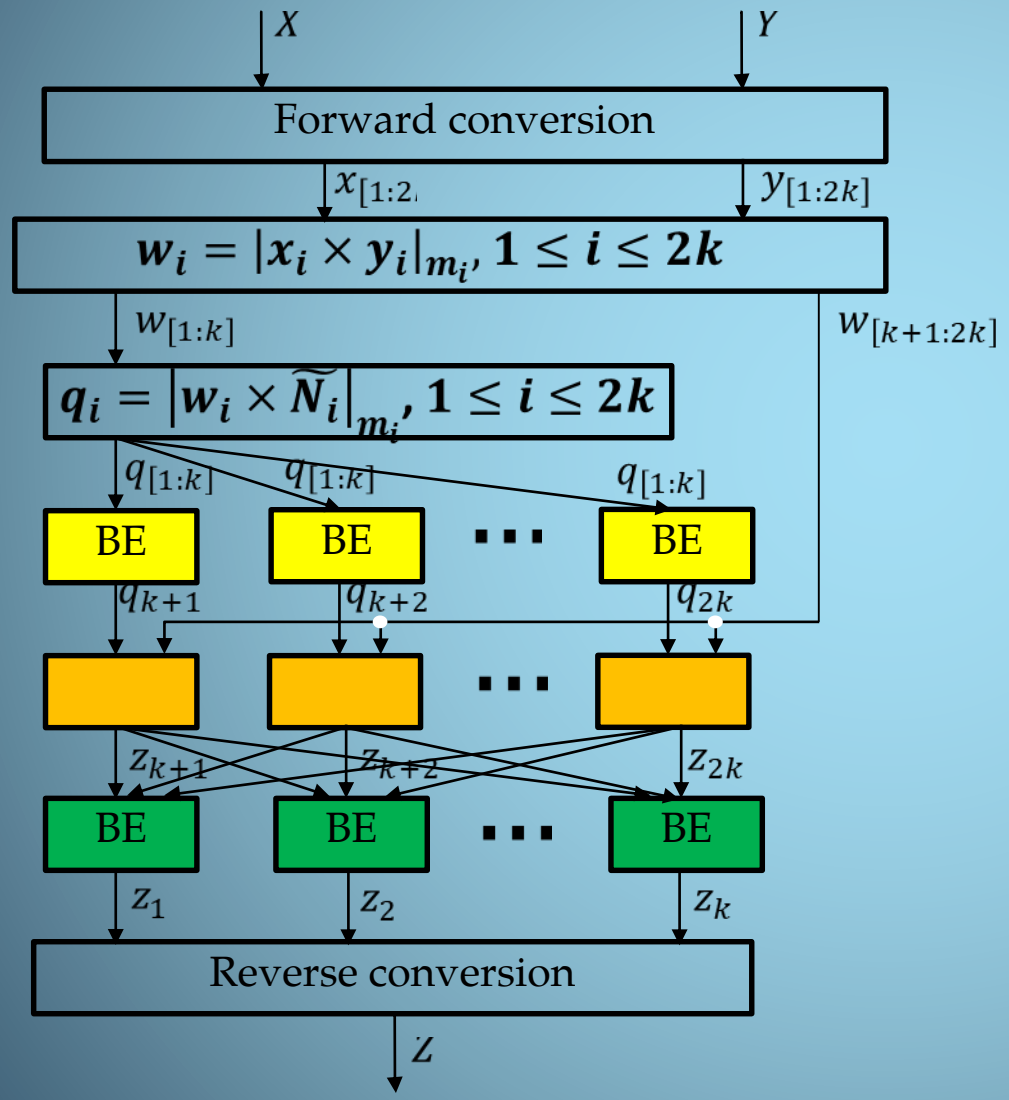
$$x_j = \left| a_k \left| \prod_{i=1}^{k-1} m_i \right|_{m_j} + \cdots \left| a_3 |m_2 m_1|_{m_j} + \left| a_2 |m_1|_{m_j} \right|_{m_j} + a_1 \right|_{m_j} \cdots \right|_{m_j}$$

$$|\gamma|_{m_x} = \left| \left(\sum_{i=1}^{k} \left| |x_i M_i^{-1}|_{m_i} M_i \right|_{m_x} - |X|_{m_x} \right) |M^{-1}|_{m_x} \right|_{m_x}, \; x_{k+j} = \left| \sum_{i=1}^{k} \left| |x_i M_i^{-1}|_{m_i} M_i \right|_{m_{k+j}} - |\gamma M|_{m_{k+j}} \right|_{m_{k+j}}$$

$$\gamma = \left\lfloor \sum_{i=1}^{k} \frac{x_i M_i^{-1}}{m_i} \right\rfloor \qquad \hat{\gamma} = \left\lfloor \sum_{i=1}^{k} \frac{trunc(\xi_i)}{2^r} + \alpha \right\rfloor$$

BE-MMM: Architecture and formulas

Schematic of RNS-MMM with two consecutive BE operations



$$q_{k+j} = BE(m_{k+j}, q_1, q_2 \dots q_k) = \left| \left(\sum_{i=1}^{k} \left| |q_i M_i^{-1}|_{m_i} M_i \right|_{m_j} \right) - |\gamma M|_{m_j} \right|_{m_j},$$

$$z_j = \left| \left(w_j + |N|_{m_j} \times q_j \right) |M^{-1}|_{m_j} \right|_{m_j},$$

$$k + 1 \le j \le 2k$$

$$z_i = BE(m_i, z_{k+1}, z_{k+2} \dots z_{2k}) = \left| \left(\sum_{j=k+1}^{2k} \left| |z_j M_j^{-1}|_{m_j} M_j \right|_{m_j} \right) - |\gamma M|_{m_i} \right|_{m_i},$$

$$1 \le i \le k$$

Motivation

- **Problem:** RNS-unfriendly division operation
- **Current Solution**: Base extension
  - BE-MMM: Two non-overlapped k-size co-prime $(2^r - \delta)$ moduli sets: Dynamic range (DR) of each set: 2^{kr}
  - BE-MMM drawbacks
    - ✓ Requires double of the essential number of co-prime moduli to support the working DR
- **Proposed Solution: SB-MMM: No base extension** $\Longrightarrow$
  - X^2 DR (i.e., 2^{2kr}) size of the same r
  - or same DR with half bit-width residue channels (i.e., $r/2$)
  - Example : $r = 24$

$$\{2^{170}, 2^{294}, 2^{506}, 2^{835}, 2^{1309}\}, \text{ for } k \in \{22,43,86,171,342\}$$

Proposed algorithm (**SB-MMM**)

Inputs: $m_i, x_i, y_i, 0 \le i \le k$, Outputs: $\widehat{z_i'}^{+}, 0 \le i \le k$

1) For $i = 0$ to k do par $\left\{ w_i = |x_i y_i|_{m_i}; \xi_{x_i} = \left|x_i M_i^{-1}\right|_{m_i}; \xi_{y_i} = \left|y_i M_i^{-1}\right|_{m_i} \right\}$;

2) For $i = 0$ to k do par $\left\{ \begin{array}{c} \xi_{\omega_i} = \left| w_i \left| \widehat{N} M_i^{-1} \right|_{m_i} \right|_{m_i}; \\ u_i = \left| w_i \left| M_i^{-1} \left(\left\lfloor \frac{M_i}{m_i} \right\rfloor M_i^{-1} + \left\lfloor \frac{N}{m_i} \right\rfloor \widetilde{N} \right) \right|_{m_i} \right|_{m_i} \end{array} \right\}$;

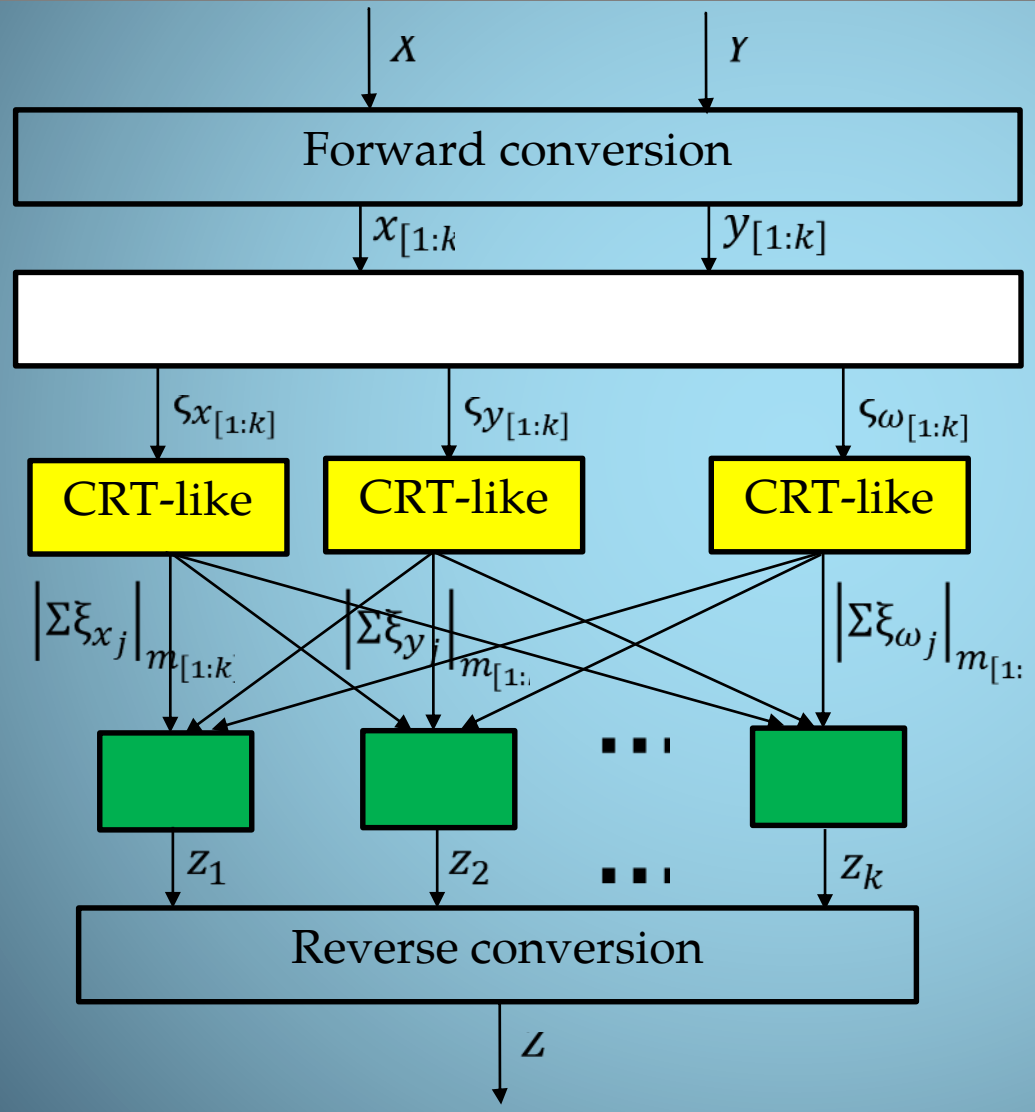
3) For $i = 0$ to k do par $\left\{ s_i = \left| \left| \xi_{x_i} \xi_{y_i} |M_i|_{m_i} m_i^{-1} + \xi_{\omega_i} |N|_{m_i} m_i^{-1} \right|_{2^{2r}} \right|_{m_i} \right\}$;

4) For $i = 0$ to k do par $\left\{ \begin{array}{l} \boldsymbol{p_{x_i} = \left(\sum_{j=0, j \neq i}^{k} \xi_{x_j} m_j^{-1} \right) - \left\lfloor \gamma_X^e + \frac{1}{2} \right\rfloor;} \\ \boldsymbol{p_{y_i} = \left(\sum_{j=0, j \neq i}^{k} \xi_{y_j} m_j^{-1} \right) - \left\lfloor \gamma_Y^e + \frac{1}{2} \right\rfloor;} \\ \boldsymbol{p_{\omega_i} = \left(\sum_{j=0, j \neq i}^{k} \xi_{\omega_j} m_j^{-1} \right) - \left\lfloor \gamma_\Omega^e + \frac{1}{2} \right\rfloor;} \end{array} \right\}$;

5) For $i = 0$ to k do par $\widehat{z_i'}^{+} = \left| N + x_i \left| p_{y_i} \right|_{m_i} + y_i \left| p_{x_i} \right|_{m_i} + |N|_{m_i} \left| p_{\omega_i} \right|_{m_i} + u_i + s_i \right|_{m_i}$; ■

SB-MMM: Architecture and formulas

Schematic of RNS-MMM with three parallel CRT-like operations, replacing the two consecutive BE operations of the conventional scheme



$$\xi_{x_i} = \left|x_i M_i^{-1}\right|_{m_i}$$

$$\xi_{y_i} = \left|y_i M_i^{-1}\right|_{m_i}$$

$$\xi_{\omega_i} = \left|x_i y_i \left|\widehat{N} M_i^{-1}\right|_{m_i}\right|_{m_i}$$

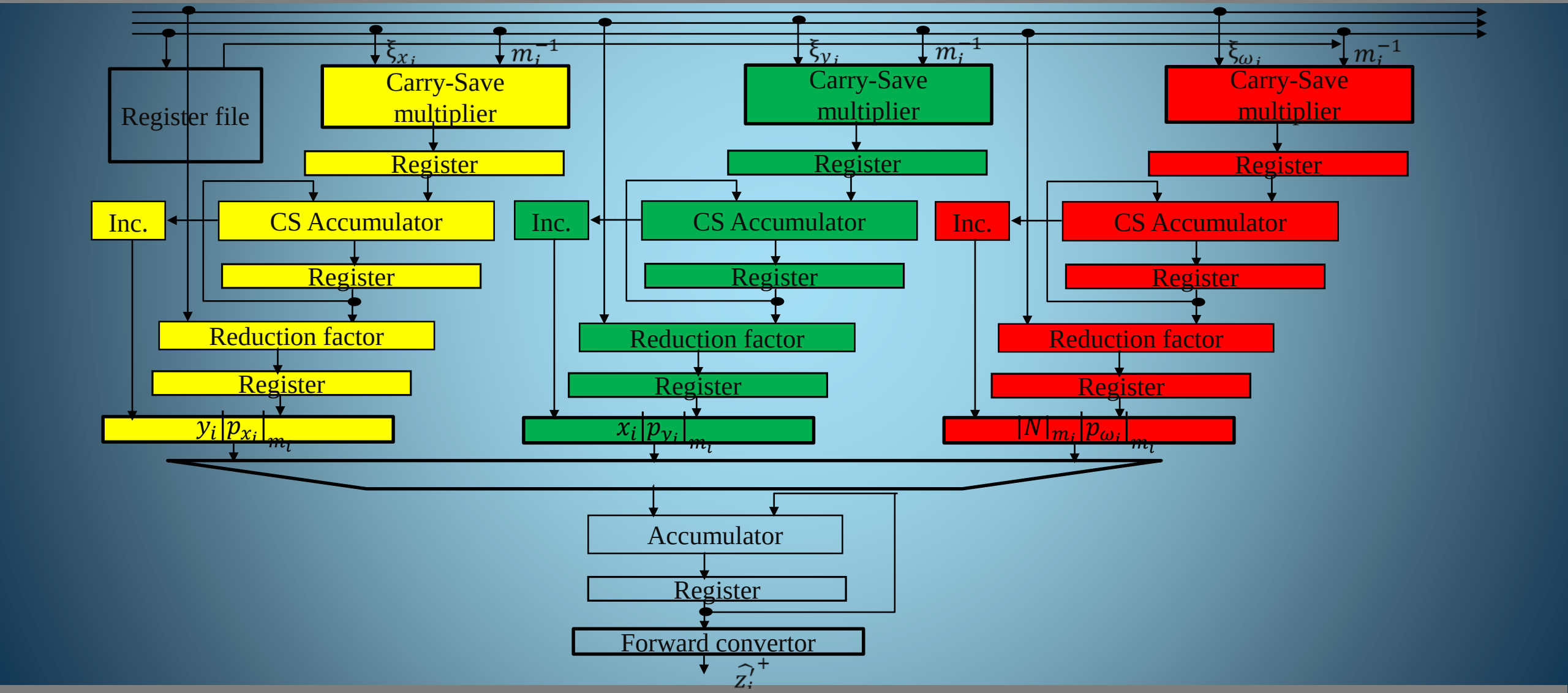
$$\left|\Sigma\xi_{x_j}\right|_{m_{[1:k]}} = \left(\sum_{j=0, j\neq i}^{k} \xi_{x_j} m_j^{-1}\right) - \left|\gamma_X^e + \frac{1}{2}\right|$$

$$\left|\Sigma\xi_{y_j}\right|_{m_{[1:k]}} = \left(\sum_{j=0, j\neq i}^{k} \xi_{y_j} m_j^{-1}\right) - \left|\gamma_Y^e + \frac{1}{2}\right|$$

$$\left|\Sigma\xi_{\omega_j}\right|_{m_{[1:k]}} = \left(\sum_{j=0, j\neq i}^{k} \xi_{\omega_j} m_j^{-1}\right) - \left|\gamma_\Omega^e + \frac{1}{2}\right|$$

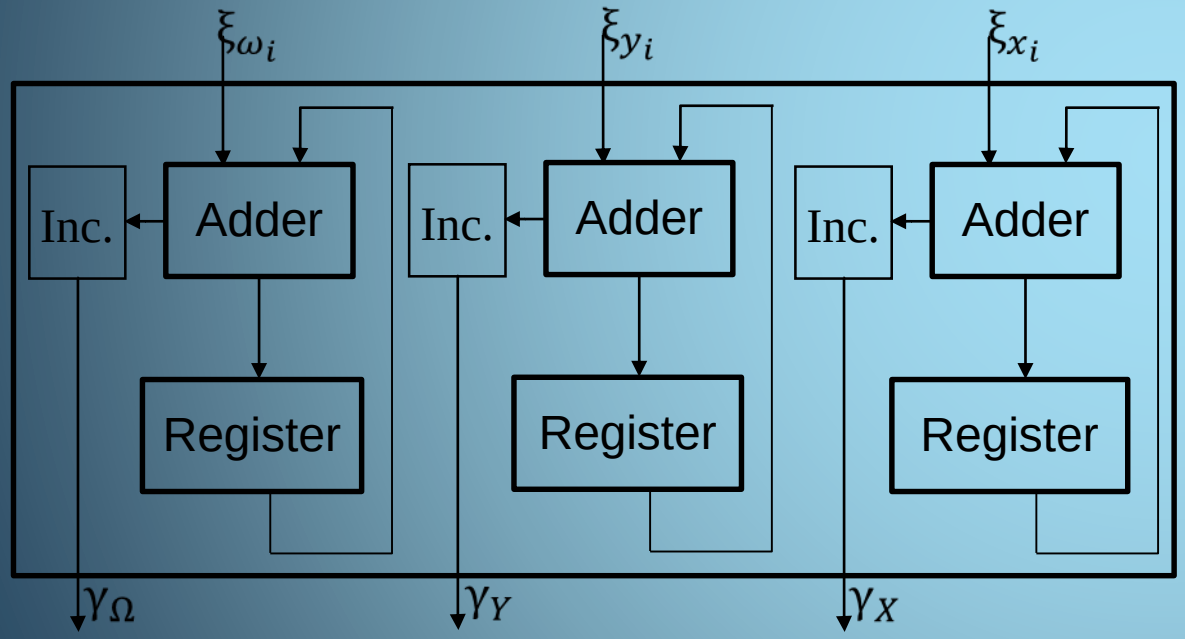
$$z_i = x_i \left|\Sigma\xi_{y_j}\right|_{m_{[1:k]}} + y_i \left|\Sigma\xi_{x_j}\right|_{m_{[1:k]}} + N \left|\Sigma\xi_{\omega_j}\right|_{m_{[1:k]}}$$
$$+ \left|\xi_{x_i}\xi_{y_i}|M_i|_{m_i} m_i^{-1} + |N|_{m_i}\xi_{\omega_i} m_i^{-1}\right|_{2^{2r}}$$
$$+ x_i y_i \left|M_i^{-1}\left(\left|\frac{M_i}{m_i}\right| M_i^{-1} + \left|\frac{N}{m_i}\right| \widetilde{N}\right)\right|_{m_i} + N$$

SB-MMM: Detailed Architecture

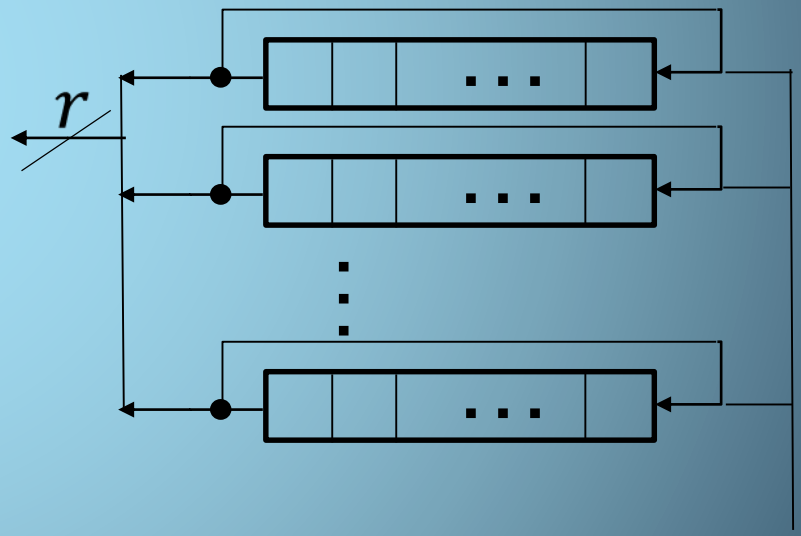


Definitive derivation of the reduction factors

Main architecture



The cyclic register file



Initial value of MAC = 2^{r-1} for supporting $\left\lfloor \boldsymbol{\gamma} + \frac{\mathbf{1}}{\mathbf{2}} \right\rfloor$

Definitive derivation via an additional modulo $m_0 = 8$ that allows $\Gamma = M > 6N$

Three scenarios for using SB-MMM

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

Example: Key-length $= 1024$, $r = 64 \Longrightarrow k = 16 \Longrightarrow$
BE-MMM delay: $2 \times 16M_{64}$ delay. Total # of moduli: 32.

$r = 32 \Longrightarrow k = 32 \Longrightarrow$ **SB-MMM** delay: $32M_{32}$ delay
($M_{32\approx}$ delay ≈80% M_{64} delay in parallel PPR and PPA).
Total # of moduli: 32.

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

Example: key-length $n = 1024$,
BE-MMM: $r = 32$: Total # of free-δ co-prime moduli of the form $2^{32} - \delta$: 384000
Total # of moduli-set selection: $\binom{384000}{32} \times \binom{383968}{32} \approx 2^{2374}$.
SB-MMM: $r = 64$. Total # of free-δ co-prime moduli of the form $2^{64} - \delta > 2^{51}$
Total # of moduli-set selection: $> \binom{2^{51}}{16} \approx 2^{816}$

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

Example: key-length $n = 1024$,
BE-MMM: $r = 32 \Longrightarrow k = 32 \Longrightarrow$ Total # of moduli: $2 \times 32 = 64 \Longrightarrow$
Delay: $2 \times 32M_{32}$
SB-MMM: $r = 32 \Longrightarrow k = 32 \Longrightarrow$ Total # of moduli: $32 \Longrightarrow$
Delay: $32M_{32}$

of multiplications: SB-MMM versus [5]

Steps of the SB-MMM algorithm	# 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] (Unpipelined)	$6k + 15$	$k(6k + 15)$
Ref. [5] (Pipelined)	$2k + 22$	

The CDP of un-pipelined SB-MMM vs. similar design of [5] is more than 83% shorter and 50% shorter than that of its pipelined version

Cost of the latter advantage is 50% increase In the total # of multiplications vs. that of [5]

Synthesis result: SB-MMM versus [5]

Design	Size of the moduli pool	Per residue channel		r	k	# of clock cycle	Clock time (ns)	Probability Of successful attacks	n	MMM Delay		AT $\left(ms \times mm^2\right)$	PDP $\left(ms \times mW\right)$
		Area (mm^2)	Power (mW)							(ns)	Ratio		
SB-MMM	1981	0.112	66.5	24	43	75	0.76	2^{-294}	1024	57	1.21	0.274	0.163
					86	118		2^{-506}	2048	90	1.45	0.866	0.514
	384000	0.135	62.3	32	32	64	0.79	2^{-369}	1024	51	1.08	0.221	0.102
					64	96		2^{-678}	2048	76	1.22	0.660	0.303
	$> 2^{51}$	0.364	127.9	64	16	48	0.97	$< 2^{-816}$	1024	47	1	0.274	0.096
					32	64		$< 2^{-1631}$	2048	62	1	0.723	0.253
[5]	251	0.079	17.4	32	32	80	1.86	2^{-134}	1024	149	3.17	0.380	0.083

Selected results: SB-MMM versus [5]

- **62%**, **66%**, and **69%** delay reductions, for channel widths $r = 24$, 32, and 64, respectively, versus $\mathrm{r} = 32$ of [5]
- **28%**, **42%**, and **28%** reduced area-time (AT) product, for $r = 24$, 32, and 64, respectively
- Reduction in the probability of successful side channel attacks by a scale of 2^{160}

Conclusion

- New RNS-MMM with three parallel CRT-like operations instead of two consecutive similar operations of the BE-MMM
- Definitive parallel derivation of three reduction factors
- 60% speedup, and 20% less area cost of proposed algorithm for key-length $n = 1024$, and channel width $r = 24$ in comparison to the best previous work [5], with $r = 32$
- Significant reduction in the probability of successful side channel attacks