RSA-CRT on a RNS/PMNS Hybrid System

DIDIER L-S **DUTOIS-RUIZ Alexy** ROBERT J-M

Université de Toulon

November 5, 2025

RSA

RSA

Main Operations

Encryption $m^e \mod N$

Decryption $c^d \mod N$

Signature $m^d \mod N$

$$\text{Where} \quad \begin{array}{l} (p,q) \in \mathbb{P}^2 \; N = pq \\ \text{GCD}\,(e, \varphi(N)) = 1 \\ d = e^{-1} \mod \varphi(n) \end{array}$$

$$\underbrace{(e, N)}_{\text{Public Key}} \qquad \underbrace{(d, p, q)}_{\text{Private Key}}$$

RSA

Main Operations

Encryption $m^e \mod N$

Decryption $c^d \mod N$

Signature $m^d \mod N$

$$\text{Where} \quad \begin{array}{l} (p,q) \in \mathbb{P}^2 \; N = pq \\ \text{GCD}\,(e, \varphi(N)) = 1 \\ d = e^{-1} \mod \varphi(n) \end{array}$$

$\underbrace{(e, N)}$ Public Key

$\underbrace{(d, p, q)}$ Private Key

Core Operation

Modular Exponentiation

RSA-CRT

Channel (thread) 1

$$d_p = |d|_{p-1}$$

$$S_p = \left| m^{d_p} \right|_p$$

Channel (thread) 2

$$d_q = |d|_{q-1}$$

$$S_q = \left| m^{d_q} \right|_q$$

Garner's Reconstruction

$$S = S_q + q \left| (S_p - S_q) \left| q^{-1} \right|_p \right|_p = \left| m^d \right|_N$$

Residue Number System (RNS)

Chinese Remainder Theorem (CRT)

$$M = \prod_{i=1}^{k} p_i$$

$$\forall i, j \in [\![0, k]\!],\ i \neq j, \mathrm{GCD}(p_i, p_j) = 1$$

$$\mathbb{Z}/M\mathbb{Z} \simeq \mathbb{Z}/p_1\mathbb{Z} \times \ldots \times \mathbb{Z}/p_k\mathbb{Z}$$

$$a \in \mathbb{Z}/M\mathbb{Z} \begin{cases} a \equiv r_1 \mod p_1 \\ \ldots \\ a \equiv r_k \mod p_k \end{cases}$$

Polynomial Modular Number System

Polynomial Modular Number System (PMNS)

Definition

$$\mathcal{B} = (n, p, \gamma, \rho, E)$$

$$\deg(E) = n, e_n = 1$$

$$\forall A \in \mathcal{B}, \deg(A) < n, \|A\|_\infty < \rho$$

Adapted Modular Number System (AMNS): $E = X^n - \lambda$

Representations

Definition

$$a \in \mathbb{Z}/p\mathbb{Z}$$

$$\mathcal{B} = (n, p, \gamma, \rho, E)$$

$$A(X) = \sum_{i=0}^{n-1} a_i X^i \in \mathcal{B}$$

$$A \underset{\mathcal{B}}{\equiv} a \iff A(\gamma) \equiv a \mod p$$

Main Operations in PMNS

- Conversions
- Multiplication
- Squaring
- Exponentiation

Main Operations in PMNS

- Conversions
- Multiplication
- Squaring
- Exponentiation

Vectorised Operations

Our implementation makes intensive usage of AVX512 vectorised instructions set.

On PMNS Multiplication's stability

$$A \in \mathcal{B} \qquad B \in \mathcal{B}$$

$$\deg(A) \leq n-1 \qquad \deg(B) \leq n-1$$

$$\widetilde{C} = A \times B$$

On PMNS Multiplication's stability

$$A \in \mathcal{B} \qquad B \in \mathcal{B}$$

$$\deg(A) \leq n-1 \qquad \deg(B) \leq n-1$$

$$\widetilde{C} = A \times B$$

$$\deg\left(\widetilde{C}\right) > n-1$$

$$\left\|\widetilde{C}\right\|_\infty > \rho$$

$$\widetilde{C} \notin \mathcal{B}$$

Reduction Operations

External Reduction

$$C' = \mathrm{ExtRed}\left(\widetilde{C}\right)$$

$$\deg\left(C'\right) \leq n - 1$$

Internal Reduction

$$C = \mathrm{IntRed}\left(C'\right)$$

$$\|C\|_\infty < \rho$$

External Reduction

$$E = \sum_{i=0}^{n} e_i X^i,\ e_n = 1$$

Definition

$$C' = \widetilde{C} \mod E$$

$$\mathcal{E} = \begin{bmatrix} -e_0 & -e_1 & \dots & -e_{n-1} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \begin{matrix} \leftarrow X^n \mod E \\ \leftarrow X^{n+1} \mod E \\ \leftarrow \dots \\ \leftarrow X^{2n-2} \mod E \end{matrix}$$

$$C' = (\widetilde{c_0}, \dots, \widetilde{c_{n-1}}) + (\widetilde{c_n}, \dots, \widetilde{c_{2n-2}})\mathcal{E}$$

External Reduction (AMNS)

$$E = X^n - \lambda \iff X^n \equiv \lambda \mod E$$

$$C' = (\widetilde{c_0}, \ldots, \widetilde{c_{n-1}}) + \lambda(\widetilde{c_n}, \ldots, \widetilde{c_{2n-2}}, 0)$$

Multiplication

deg	**0**	**1**	**2**	...	**n − 3**	**n − 2**	**n − 1**
	$a_{n-1}b_1$	$a_{n-1}b_2$	$a_{n-1}b_3$	...	$a_{n-1}b_{n-2}$	$a_{n-1}b_{n-1}$	$a_{n-1}b_0$
+	$a_{n-2}b_2$	$a_{n-2}b_3$	$a_{n-2}b_4$	...	$a_{n-2}b_{n-1}$	$a_{n-2}b_0$	$a_{n-2}b_1$
+	$a_{n-3}b_3$	$a_{n-3}b_4$	$a_{n-3}b_5$	...	$a_{n-3}b_0$	$a_{n-3}b_1$	$a_{n-3}b_2$
+	...	...	...	...	...	...	...
+	a_1b_{n-1}	a_1b_0	a_1b_1	...	a_1b_{n-4}	a_1b_{n-3}	a_1b_{n-2}
+	a_0b_0	a_0b_1	a_0b_2	...	a_0b_{n-3}	a_0b_{n-2}	a_0b_{n-1}

Table: Partial products scheme

Multiplication

deg	**0**	**1**	**2**	...	**n − 3**	**n − 2**	**n − 1**
	$a_{n-1}b_1$	$a_{n-1}b_2$	$a_{n-1}b_3$	...	$a_{n-1}b_{n-2}$	$a_{n-1}b_{n-1}$	$a_{n-1}b_0$
+	$a_{n-2}b_2$	$a_{n-2}b_3$	$a_{n-2}b_4$	...	$a_{n-2}b_{n-1}$	$a_{n-2}b_0$	$a_{n-2}b_1$
+	$a_{n-3}b_3$	$a_{n-3}b_4$	$a_{n-3}b_5$	...	$a_{n-3}b_0$	$a_{n-3}b_1$	$a_{n-3}b_2$
+	...	...	...	...	...	...	...
+	a_1b_{n-1}	a_1b_0	a_1b_1	...	a_1b_{n-4}	a_1b_{n-3}	a_1b_{n-2}
+	a_0b_0	a_0b_1	a_0b_2	...	a_0b_{n-3}	a_0b_{n-2}	a_0b_{n-1}

Table: Partial products scheme

External and Internal Reductions

$$E = X^n - \lambda \Longleftrightarrow X^n \equiv \lambda \mod E$$

External Reduction

$$\text{acc512} \leftarrow \text{acc512} + (\lambda - 1) \times \text{cx512}$$

Internal Reduction

$$\text{acc512} \leftarrow \text{IntRed}(\text{acc512})$$

Modular Exponentiation

Objective

$$A \underset{\mathcal{B}}{\equiv} a$$

$$A^e \underset{\mathcal{B}}{\equiv} a^e$$

Fixed Window Exponentiation

Fixed Window Exponentiation

Moving window

$$e = \overbrace{\boxed{0\ldots0e_k e_{k-1}}}^{w \text{ bits}} \ldots \boxed{e_{11}e_{10}e_9e_8}\boxed{e_7e_6e_5e_4}\boxed{e_3e_2e_1e_0}$$

$\rightarrow$ $\rightarrow$ $\rightarrow$

Multipliers computation

$$s = 2^w$$

$$\text{Multipliers} = \left(\mathbf{1}_{\mathcal{B}}, A, A^2, \ldots, A^{s-1}\right)$$

Algorithm: Exp

Require: $A \underset{\mathcal{B}}{\equiv} a$, $e \in \mathbb{N}$, $w \in \mathbb{N}$, $s = 2^w$

Ensure: $R \underset{\mathcal{B}}{\equiv} a^e$

1: $R \leftarrow \mathbf{1}_{\mathcal{B}}$
2: **for** t the next w bits of e **do**
3: **for** $i = 1, \ldots, s$ **do**
4: $R \leftarrow R^2$
5: **end for**
6: $M \leftarrow \textsc{Select}(Multipliers, s, t)$ ▷ $M[t]$ Constant time
7: $R \leftarrow R \times M$
8: **end for**
9: **return** R

Constant Time Selection

Timing attack

Naturally accessing an element in an array is not a constant time operation

Algorithm: Select

Require: Multipliers: A s-length polynomials array in $\mathcal{B}$, $s = 2^w$,
$t \in [\![0, s-1]\!]$
Ensure: $M = \text{Multipliers}[t]$

1: $M \leftarrow \mathbf{0}_{\mathcal{B}}$
2: **for** $i = 0, \ldots, s-1$ **do**
3: $\quad \text{Mask}[i] \leftarrow i = t$ ▷ 1 if $i = t$, 0 else
4: **end for**
5: **for** $k = 0, \ldots, s-1$ **do**
6: $\quad M \leftarrow M$ **or** $\text{Multipliers}[k] \times \text{Mask}[k]$ ▷ Bitwise OR
7: **end for**
8: **return** M

Contributions

Squaring

Objectif

$$A \underset{\mathcal{B}}{\equiv} a$$

$$A^2 \underset{\mathcal{B}}{\equiv} a^2$$

Squaring

deg	**0**	**1**	**2**	**3**	...	$\mathbf{n-2}$	$\mathbf{n-1}$	**n**	$\mathbf{n+1}$	...	$\mathbf{2n-3}$	$\mathbf{2n-2}$
	a_0^2	a_0a_1	a_0a_2	a_0a_3	...	a_0a_{n-2}	a_0a_{n-1}					
+		a_1a_0	a_1^2	a_1a_2	...	a_1a_{n-3}	a_1a_{n-2}	a_1a_{n-1}				
+			a_2a_0	a_2a_1	...	a_2a_{n-4}	a_2a_{n-3}	a_2a_{n-2}	a_2a_{n-1}			
+					...	...	...	...	...	...		
+						$a_{n-2}a_0$	$a_{n-2}a_1$	$a_{n-2}a_2$	$a_{n-2}a_4$	...	$a_{n-2}a_{n-1}$	
+							$a_{n-1}a_0$	$a_{n-1}a_1$	$a_{n-1}a_2$	...	$a_{n-1}a_{n-2}$	a_{n-1}^2

Figure: Partial products scheme

Squaring

deg	0	1	2	3	...	n − 2	n − 1	n	n + 1	...	2n − 3	2n − 2
	a_0^2	a_0a_1	a_0a_2	a_0a_3	...	a_0a_{n-2}	a_0a_{n-1}					
+		~~a_1a_0~~	a_1^2	a_1a_2	...	a_1a_{n-3}	a_1a_{n-2}	~~a_1a_{n-1}~~				
+			~~a_2a_0~~	~~a_2a_1~~	...	a_2a_{n-4}	a_2a_{n-3}	~~a_2a_{n-2}~~	~~a_2a_{n-1}~~			
+					...	...	...	...	...	...		
+						~~$a_{n-2}a_0$~~	~~$a_{n-2}a_1$~~	$a_{n-2}a_2$	$a_{n-2}a_4$	...	~~$a_{n-2}a_{n-1}$~~	
+							~~$a_{n-1}a_0$~~	$a_{n-1}a_1$	$a_{n-1}a_2$	...	$a_{n-1}a_{n-2}$	a_{n-1}^2

Figure: Partial products scheme

Doubled coefficients

Low part

$$
\begin{array}{rccl}
\text{acc512} \leftarrow & a_0 & \times & (0, a_1, a_2, a_3 \ldots, a_{n-2}, a_{n-1}) \\
+ & a_1 & \times & (0, 0, 0, a_2 \ldots, a_{n-3}, a_{n-2}) \\
+ & \ldots & \ldots & \ldots \\
+ & a_{\lfloor \frac{n}{2}-1 \rfloor} & \times & \ldots
\end{array}
$$

High part

$$
\begin{array}{rccl}
\text{cx512} \leftarrow & a_{n-1} & \times & (a_1, a_2 \ldots, a_{n-3}, a_{n-2}, a_{n-1}, 0) \\
+ & a_{n-2} & \times & (a_2, a_3, \ldots, a_{n-2}, 0, 0, 0) \\
+ & \ldots & \ldots & \ldots \\
+ & a_{\lfloor \frac{n}{2} \rfloor} & \times & \ldots
\end{array}
$$

Squared coefficients

High part

$$\text{acc512} \leftarrow 2 \times \text{acc512} + \left(a_0^2, 0, a_1^2, \ldots, a_{\lfloor \frac{n}{2} - 1 \rfloor}^2 \right)$$

Low part

$$\text{cx512} \leftarrow 2 \times \text{cx512} + \left(a_{\lfloor \frac{n}{2} \rfloor}^2, 0, \ldots, a_{n-1}^2 \right)$$

External and Internal Reductions

$$E = X^n - \lambda \Longleftrightarrow X^n \equiv \lambda \mod E$$

External Reduction

$$\text{acc512} \leftarrow \text{acc512} + \lambda \times \text{cx512}$$

Internal Reduction

$\text{acc512} \leftarrow \text{IntRed}(\text{acc512})$

RSA-CRT multiprime

$$N = \prod_{j=0}^{k} p_j$$

Objective

- Increasing the number of channels
- Decresing the size of each prime p_i

$$N_i = \prod_{\substack{j=0 \\ j \neq i}}^{k} p_j$$

$$N_i' = N_i^{-1} \mod p_i$$

RSA-CRT multiprime

Computation in each channel (thread)

$$d_i = |d|_{p_i - 1}$$

$$S_i = \left| m^{d_i} \right|_{p_i}$$

Reconstruction (CRT)

$$S = \left| \sum_{i=1}^{k} S_i N_i N_i' \right|_N$$

Performances

RSA-CRT

Number of channels $\times$ size of p_i	Implementation	Single	2 Threads	4 Threads	8 Threads
2×2048	PMNS (This work)	10220712	5121905		
	GMP	16546242	8635423		
	OpenSSL	13208496	6642533		
	OpenSSL ($\times 2$)	**4997144**			
4×1024	PMNS (This work)	2600118	1351519	**708477**	
	GMP	4509863	2862107	1983391	
	OpenSSL	3708709	1905005	1003183	
	OpenSSL ($\times 2$)	**1826561**	**966330**		
8×512	PMNS (This work)	**1125524**	**585619**	**315885**	**188722**
	GMP	1363780	924330	655177	653013
	OpenSSL	1177215	665751	410403	278538
	OpenSSL ($\times 2$)	1176203	665318	403742	

Table: Number of cycles during a RSA 4096 signature

Perspectives

State of vectorized PMNS

This work	2022	Classic Montgomery
State of the Art	2025	Multiprecision coefficients, Montgomery CIOS

Future work

- Common operand multiplication
- Truncated Montgomery

Objectives

Compete with batch computations

Thank you