21st IEEE Symposium on Computer Arithmetic (ARITH'21)

Austin, Texas, USA, April 7-10, 2013

Speaker : **Mourad Gouicem***

Fault Detection in RNS Montgomery Modular Multiplication

Jean-Claude Bajard*, Julien Eynard*, Filippo Gandino†

*LIP6 CNRS - Univ. Paris 6, France
†Politecnico di Torino, Italy

April 9th, 2013

Motivation

Using Residue Number Systems in cryptosystems ?

- Efficiency of RNS arithmetic for RSA, ECC and pairings...

  N. Guillermin *Implémentation matérielle de coprocesseurs haute performance pour la cryptographie asymétrique*. PhD. thesis, Univ. Rennes 1, 2012.

- ...on several architectures (FGPA, GPU).

  S. Antao, J.-C. Bajard, L. Sousa. *RNS-Based Elliptic Curve Point Multiplication for Massive Parallel Architectures*. The Computer Journal, Oxford University Press, 2012.

  R. Cheung et al. *FPGA implementation of pairings using residue number system and lazy reduction*. Lecture Notes in Computer Science, Springer, 2011.

- Creation of a Leak Resistant Arithmetic (LRA) based on RNS.

  J.-C. Bajard, L. Imbert, P.-Y. Liardet, Y. Teglia. *Leak Resistant Arithmetic*. Lecture Notes in Computer Science, Springer, 2004.

Goal

To exploit particularities of RNS to construct an efficient arithmetic for cryptographic applications.
$\rightarrow$ **So, what about protection of the RNS modular multiplication against fault injection attacks ?**

About RNS - Residue Number Systems

Chinese Remainder Theorem (CRT)

Let $m_1, \ldots, m_n$ be coprime integers, $M := m_1 \ldots m_n$.
Then $\mathbb{Z}/M\mathbb{Z}$ is isomorphic to $\mathbb{Z}/m_1\mathbb{Z} \times \ldots \times \mathbb{Z}/m_n\mathbb{Z}$.

Definition

- $\{m_1, \ldots, m_n\}$ is a "RNS base".
- $[\![0, M[\![$ = usual "dynamic range" ; $\mathbb{Z}/m_i\mathbb{Z}$ = "a channel".
- For $x \in [\![0, M[\![$, $x_i = |x|_{m_i} = x \bmod m_i$ is the i^{th} residue of x.

- Addition, subtraction, multiplication and exact division are performed in each channel.
- No carry propagation → indepency between channels.
- **But**, RNS = no positional number system → comparison ? modular reduction ? computations in $\mathbb{Z}/P\mathbb{Z}$?

About RNS - Modular multiplication

Classical Montgomery modular multiplication : $a \times b \bmod p$

Montgomery's technique : to choose an integer M such that division and modular reduction by M are easy ! (e.g. $M = 2^k$)

Algorithm 1 Montgomery reduction

Require: p, M, such that $gcd(p, M) = 1$ and $ab < Mp$

1. $q \leftarrow \left|-abp^{-1}\right|_M$
2. $s \leftarrow \frac{ab+qp}{M}$

return $s < 2p$, $s \equiv abM^{-1} \bmod p$

Adaptation to RNS

q easy to compute in RNS base $\mathcal{B}$ ($\rightsquigarrow M$). But division by M ?
Solution : auxiliary base $\mathcal{B}'$ coprime to $\mathcal{B}$.

J.-C. Bajard., L.-S. Didier, P. Kornerup *An RNS Montgomery Modular Multiplication Algorithm*. IEEE Transac. on Comp., 1998.

J.-C. Bajard., L.-S. Didier, P. Kornerup *Modular Multiplication, and Base Extension in Residue Number Systems*. ARITH15, 15th IEEE symposium on computer arithmetic, 2001.

About RNS - Modular multiplication

Overview of the RNS algorithm

in base $\mathcal{B}$ (mod M)	base conversion	in base $\mathcal{B}'$ (mod M')
$q = -abp^{-1}$		-
q	$\Rightarrow$	q
- (0)		$t = ab + qp$
- (?)		$s = tM^{-1}$
s	$\Leftarrow$	s

About RNS - Base conversions

Based on the CRT

Given $x_1, \ldots, x_n$, $M_i := M/m_i$, $\xi_i := \left|x_i M_i^{-1}\right|_{m_i}$,

$$x = \left|\sum_{i=1}^{n} \xi_i M_i\right|_M = \sum_{i=1}^{n} \xi_i M_i - k_x M$$

$\rightarrow$ Computation of $k_x = \lfloor \sum_{i=1}^{n} \frac{\xi_i}{m_i} \rfloor < n$?

- Shenoy and Kumaresan (89) : by adding an extra channel $m_{sk} \geqslant n$ so that $|k_x|_{m_{sk}} = k_x$. Requires to know $|x|_{m_{sk}}$.
- Bajard, Didier, Muller (97), Kawamura et al (00) : approx. $\lfloor \sum_{i=1}^{n} \frac{trunc(\xi_i)}{2^r} \rfloor$, where $2^{r-1} < m_i < 2^r$ for all i. Computed by a unit called "Cox".

Main base conversion techniques

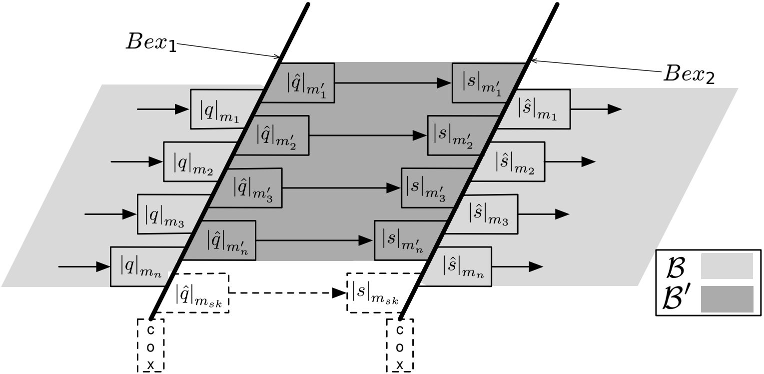
Based on the associated Mixed Radix System (MRS)

Associated MRS : $\{1, m_1, m_1 m_2, \ldots, m_1 m_2 \ldots m_{n-1}\}$
From $x_1, \ldots, x_n$, MRS coef. of x are :

$$\begin{aligned}
\tilde{x}_1 &= x_1 \\
\tilde{x}_2 &= \left|(x_2 - \tilde{x}_1)\, m_1^{-1}\right|_{m_2} \\
\tilde{x}_n &= \left|\left(\ldots (x_n - \tilde{x}_1)\, m_1^{-1} - \ldots - \tilde{x}_{n-1}\right) m_{n-1}^{-1}\right|_{m_n}
\end{aligned}$$

$x = \tilde{x}_1 + \tilde{x}_2 m_1 + \ldots + \tilde{x}_n m_1 \ldots m_{n-1}$

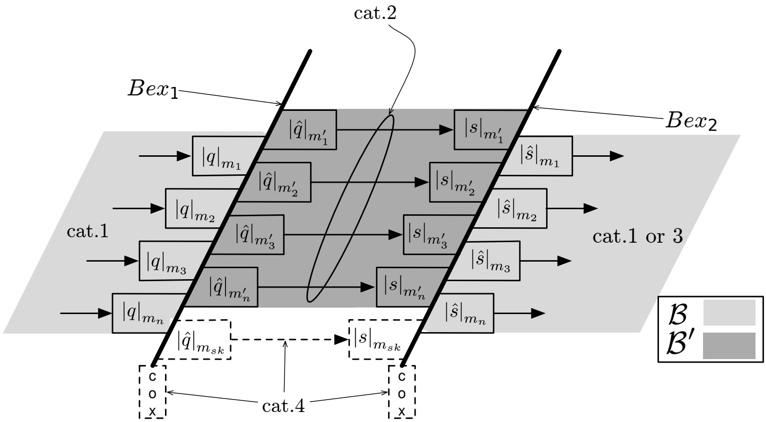
Which faults?



Which faults ?

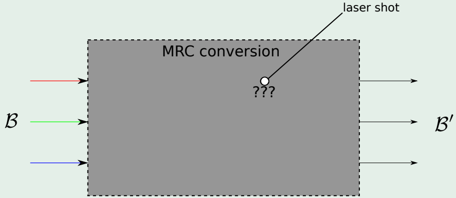
Locality condition

Practically, alteration of few bits (e.g. laser shot) ⇒ focus on one channel.



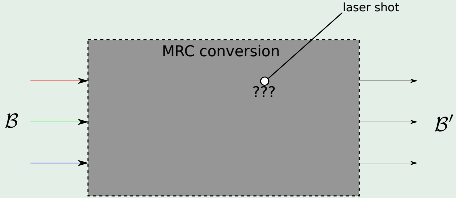
Which faults ?

What if a fault during a base conversion ?

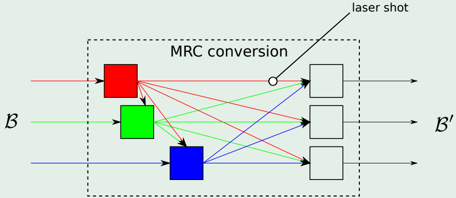


Which faults ?

What if a fault during a base conversion ?



During the 3 types of conversion : computations only in channels. E.g. :



Localized fault during a base conversion = single fault in $\mathcal{B}$ or in $\mathcal{B}'$.

Fault model

Formalisation

Theoretically, fault in a ring $\mathbb{Z}/m\mathbb{Z}$ (i.e. a single channel).

$$\left(x_1, \ldots, x_{i-1}, \left|x_i + e_i\right|_{m_i}, x_{i+1}, \ldots, x_n\right) \rightarrow \overline{x} = x + a_i M_i \in [\![0, M[\![,$$
$$a_i \in]-m_i, m_i[.$$

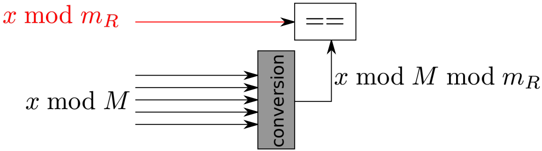
Redundant RNS and base conversion enable to detect such faults.

Redundant RNS and fault detection

- Redundant modulus m_R : $[\![0, M[\![\rightsquigarrow [\![0, m_R M[\![$.
- Single fault : $\overline{x} = x + a_i M_i \mathbf{m_R}$.
- $m_R > m_i$ and $m_R \wedge M \Rightarrow \overline{x} \in [\![M, m_R M[\![$.
  $\rightarrow [\![0, M[\![=$ correct values ; $[\![M, m_R M[\![=$ incorrect values

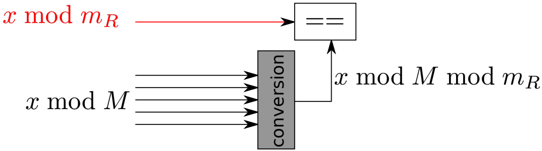
Redundant RNS and fault detection

- Redundant modulus m_R : $[\![0, M[\![\rightsquigarrow [\![0, m_R M[\![$.
- Single fault : $\overline{x} = x + a_i M_i \mathbf{m_R}$.
- $m_R > m_i$ and $m_R \wedge M \Rightarrow \overline{x} \in [\![M, m_R M[\![$.
  $\rightarrow [\![0, M[\![=$ correct values ; $[\![M, m_R M[\![=$ incorrect values
- *Consistency check* :



Redundant RNS and fault detection

- Redundant modulus m_R : $[\![0, M[\![\rightsquigarrow [\![0, m_R M[\![$.
- Single fault : $\overline{x} = x + a_i M_i \mathbf{m_R}$.
- $m_R > m_i$ and $m_R \wedge M \Rightarrow \overline{x} \in [\![M, m_R M[\![$.
  $\rightarrow [\![0, M[\![=$ correct values ; $[\![M, m_R M[\![=$ incorrect values
- *Consistency check* :



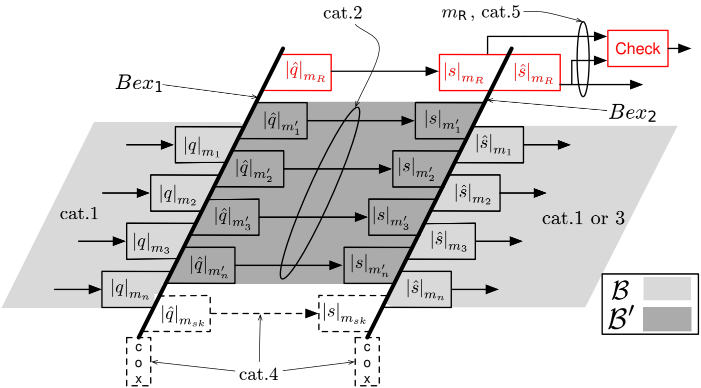
- Already known with MRC based checks.
- Proven : works with CRT based checks.

Redundant RNS modular multiplication ?

$\rightarrow$ Beware ! Base conversion = costly.

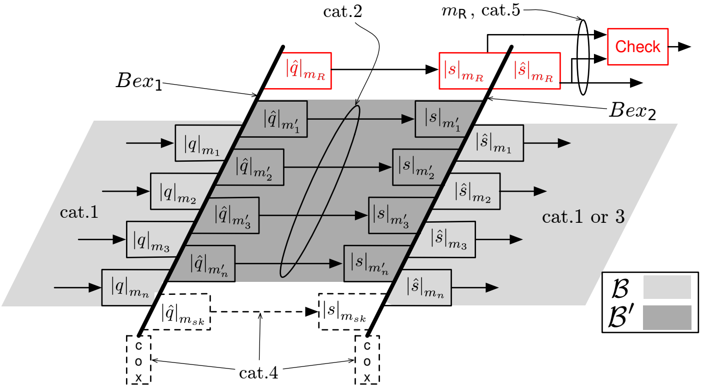
The proposed algorithm

main base $\mathcal{B}$ (mod M)	base conversion/extension	auxiliary base $\mathcal{B}'$ (mod M')	redundant channel (mod m_R)
$q = -abp^{-1}$		-	-
q	$\text{Bex}_1(q) \Rightarrow$	q	q
- (0)		$t = ab + qp$	$ab + qp$
- (?)		$s = tM^{-1}$	$(ab + qp)M^{-1}$
s	$\Leftarrow \text{Bex}_2(s \bmod M')$	s	$-$



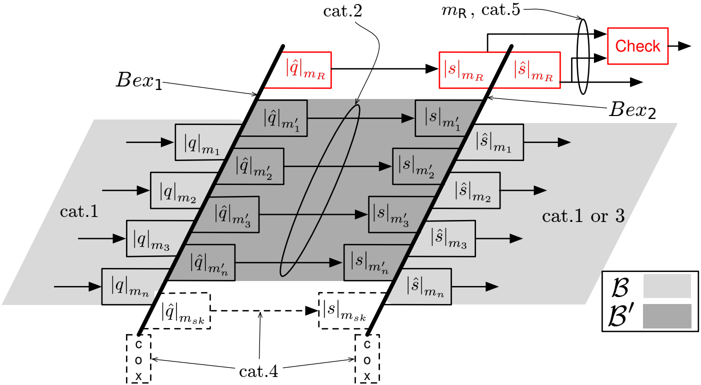
The proposed algorithm - Analysis of detection capability

Cat. 2 : Integrity of $s \bmod m_R M'$ $\rightarrow$ consistency check based on Bex_2 ?
Yes : $s = \frac{t}{M} < M'$ and so $|s|_{m_R} = tM^{-1} \bmod m_R$ computable.



The proposed algorithm - Analysis of detection capability

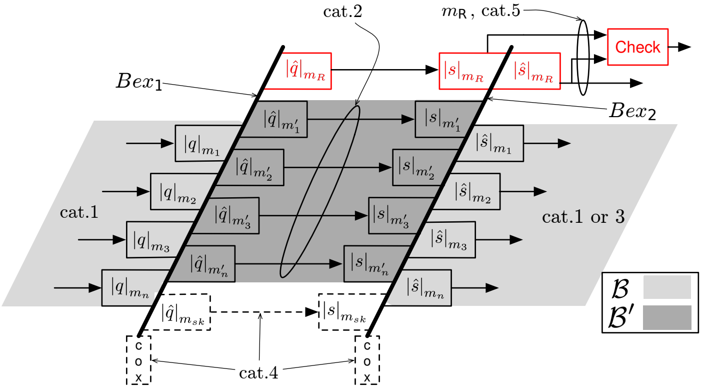
Cat. 3 : $\rightsquigarrow$ cat. 1, or needs extra consistency check



The proposed algorithm - Analysis of detection capability

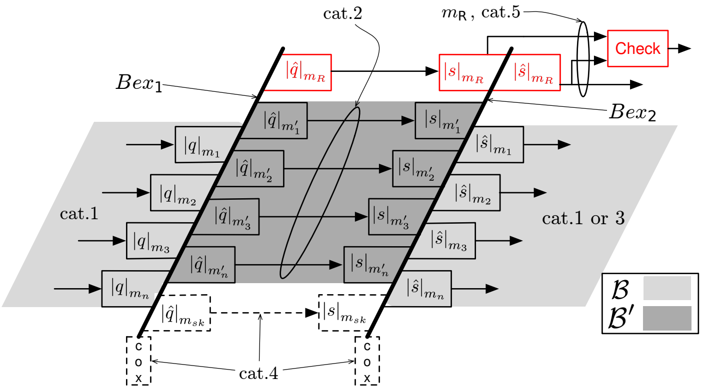
Cat. 4 : (i.e. on extra stuff for CRT based conversions)

- in Cox unit : larger bases, or two (little) Cox units...
- in m_{sk} channel : works → as category 2.

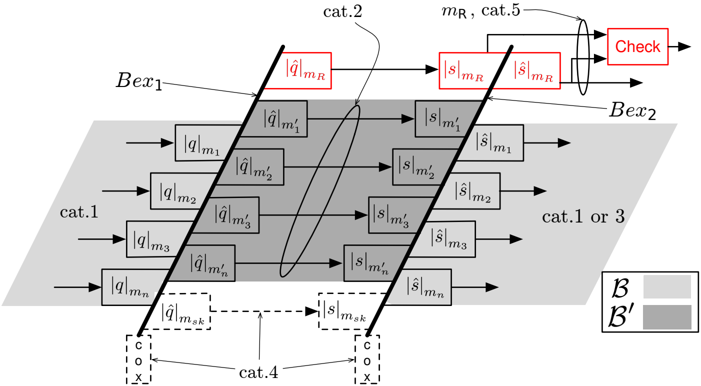


The proposed algorithm - Analysis of detection capability

Cat. 5 : obvious...



Cat. 1 : Computation of $|q|_{m_R}$ before Bex$_1$? **Impossible**.
1 fault on $q \Rightarrow$ many faults on s mod M'... **No detection ? !**



The proposed algorithm - Detection of faults of category 1

$$\overline{q_i} \Rightarrow \overline{t} = ab + \text{Bex}_1(\overline{q})p < MM' \Rightarrow \overline{t} = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overbrace{\overline{t'_1}, .., \overline{t'_n}}^{\mathcal{B}'})).$$

The proposed algorithm - Detection of faults of category 1

$$\overline{q_i} \Rightarrow \overline{t} = ab + \mathrm{Bex}_1(\overline{q})p < MM' \Rightarrow \overline{t} = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n})).$$

Consequences in redundant channel

$$\begin{aligned} \overline{t} < MM' \quad &\Rightarrow \quad \overline{t} \bmod m_R = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n})) \bmod m_R \\ &\Rightarrow \quad \overline{s} \bmod m_R = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n}))M^{-1} \bmod m_R \end{aligned}$$

The proposed algorithm - Detection of faults of category 1

$$\overline{q_i} \Rightarrow \overline{t} = ab + \text{Bex}_1(\overline{q})p < MM' \Rightarrow \overline{t} = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n})).$$

Consequences in redundant channel

$$\begin{aligned} \overline{t} < MM' \quad \Rightarrow \quad & \overline{t} \bmod m_R = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n})) \bmod m_R \\ \Rightarrow \quad & \overline{s} \bmod m_R = ((0, .., 0, \overset{\mathcal{B}}{e_i}, 0, .., 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n}))M^{-1} \bmod m_R \end{aligned}$$

Value computed by Bex_2

$$(\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n})\ M^{-1} \bmod M' \bmod m_R = ((0, \overset{\mathcal{B}}{\ldots}, 0), (\overline{t'_1}, \overset{\mathcal{B}'}{..}, \overline{t'_n}))M^{-1} \bmod m_R$$

The proposed algorithm - Detection of faults of category 1

$$\overline{q_i} \Rightarrow \overline{t} = ab + \text{Bex}_1(\overline{q})p < MM' \Rightarrow \overline{t} = ((0,..,0,\overset{\mathcal{B}}{e_i},0,..,0),(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n})).$$

Consequences in redundant channel

$$\begin{aligned} \overline{t} < MM' \quad \Rightarrow \quad & \overline{t} \bmod m_R = ((0,..,0,\overset{\mathcal{B}}{e_i},0,..,0),(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n})) \bmod m_R \\ \Rightarrow \quad & \overline{s} \bmod m_R = ((0,..,0,\overset{\mathcal{B}}{e_i},0,..,0),(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n}))M^{-1} \bmod m_R \end{aligned}$$

Value computed by Bex_2

$$(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n})\ M^{-1} \bmod M' \bmod m_R = ((0,\overset{\mathcal{B}}{\ldots},0),(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n}))M^{-1} \bmod m_R$$

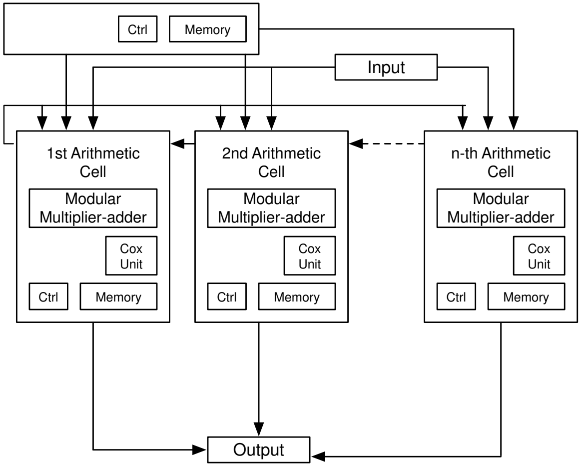
Consistency check :

$((0,..,0,\overset{\mathcal{B}}{e_i},0,..,0),(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n})) \overset{?}{\equiv} ((0,\overset{\mathcal{B}}{\ldots},0),(\overline{t'_1},\overset{\mathcal{B}'}{..},\overline{t'_n})) \bmod m_R \rightarrow$ single fault model ! It works !

An architecture

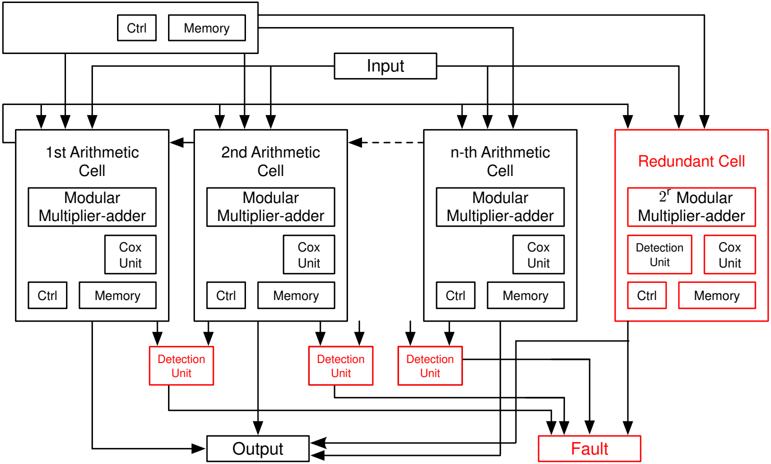
Practically, RNS Montgomery + CRT based conversions with Cox unit.

H. Nozaki, M. Motoyama, A. Shimbo, and S. Kawamura. *Implementation of RSA algorithm based on RNS Montgomery multiplication*. CHES, 2001.



An architecture

Practically, RNS Montgomery + CRT based conversions with Cox unit.



An architecture

Some informations

- Adapt fault model to size of output registers :
  $2^{r-1} < m_i < 2^r \Rightarrow m_R \geqslant 2^r$.
- Area(Detection units + Redondant cell) $\leqslant$ Area(Standard cell)
- Time cost during normal work flow : none
- Extra time cost for detection of cat. 3 faults :
  - for 1 mod. mult. $\sim 1/2$
  - $\rightarrow$ for 1 mod. exp. with Montgomery ladder $\sim 1/2 \log_2(exponent)$

Comparison to state-of-the-art

Guillermin's technique

N. Guillermin *A coprocessor for secure and high speed modular arithmetic.* Cryptology ePrint Archive, 2011.

- Specific to Cox-Rower architecture (modified Cox).
- Not compliant with LRA.
- $+ \geqslant 1$ extra not redundant channel.
- Several faults ? Hard...

Our technique

- Genericity.
- Compliant with LRA.
- $+1$ extra redundant channel. (extra area $\leqslant$ Guillermin's one)
- Several faults ? Easy !

E.g. : RSA-CRT 1024 with Montgomery ladder $\rightarrow 2 \times 1024$ mod. mult.
Guillermin : $+5\%$, us : $+1/(2 \times 1024) \sim 0.05\%$.

Conclusion

The proposed redundant RNS Montgomery multiplication algorithm :

- Genericity
- Time cost during normal work flow : none
- Time cost... just an extra (optional) final base conversion
- Efficiency
- Compliant with a Leak Resistant Arithmetic
- Adaptable to detection of several faults
- Adaptable to RNS Montgomery multiplication in $GF(p^k)$

Thank You !

Questions ?

jean-claude.bajard@lip6.fr
julien.eynard@lip6.fr
filippo.gandino@polito.it