

Computer Arithmetic for Cryptography in the Arith Group

Arnaud Tisserand

LIRMM, CNRS–Univ. Montpellier 2
Arith Group

Crypto'Puces
Porquerolles, April 16–18, 2007



Introduction

- LIRMM Laboratory
- Arith Group
- Computer Arithmetic
- Some Research Activities

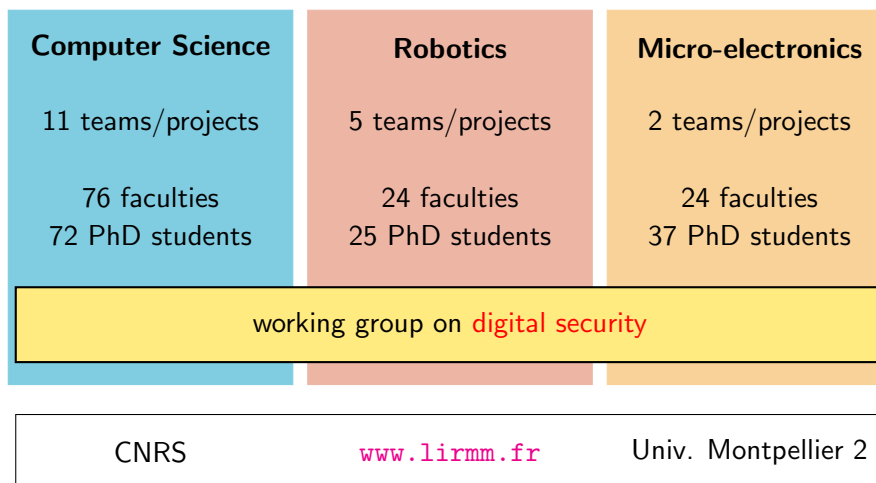
Examples

- Residue Number Systems (RNS)
- Addition Chains
- Double-Base Number Systems (DBNS)
- Library

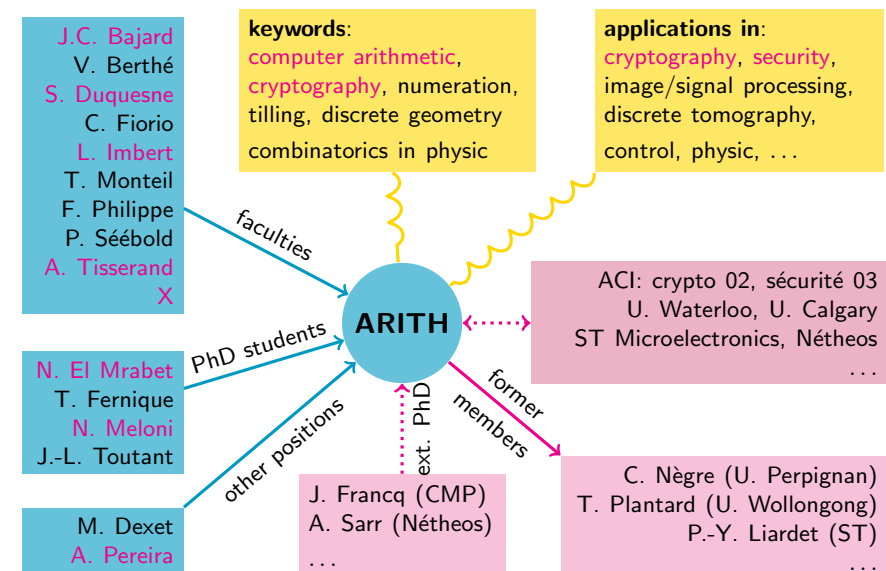
Future Prospects

LIRMM

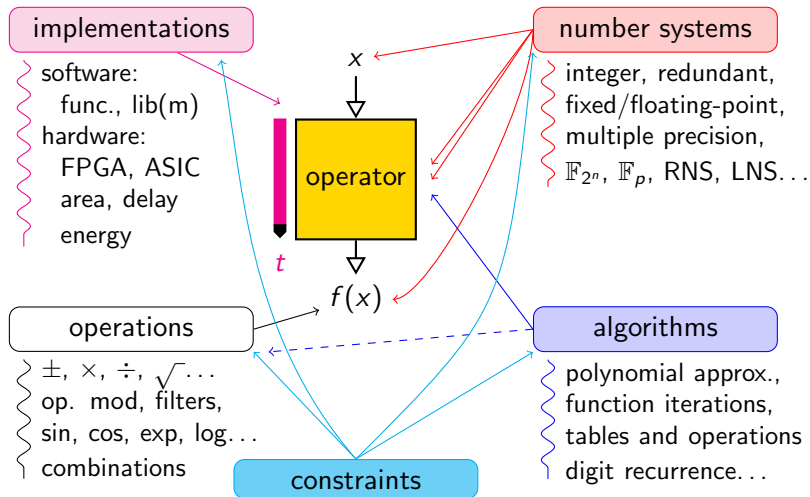
Montpellier Laboratory of Computer Science, Robotics, and Microelectronics



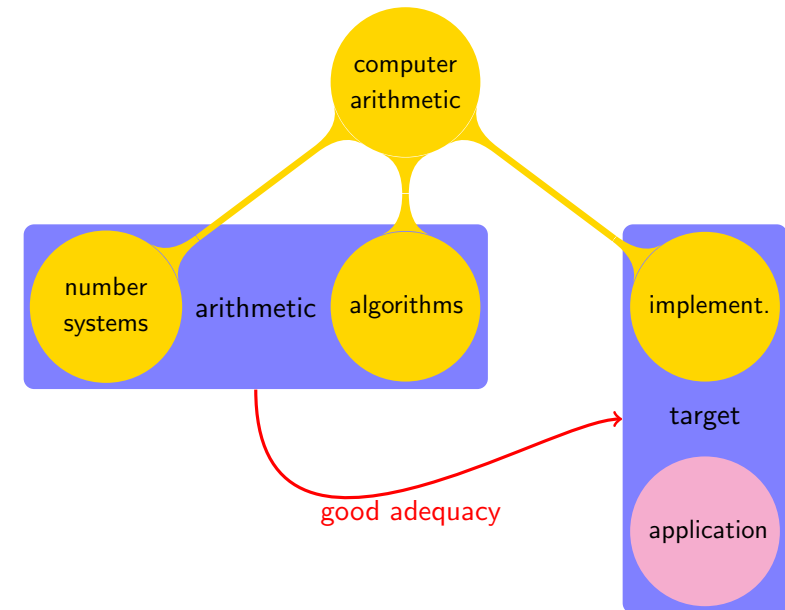
Arith Group



Computer Arithmetic



Problems in Computer Arithmetic



Practical Problems in Computer Arithmetic

- **limited support in design tools**
 - software: integer, floating-point, libraries
 - hardware: integer, fixed-point, a few IP blocs
- **validation**
 - verification of the correctness of a program (function, library, hardware bloc, circuit) at design time
- **test**
 - verification of the correctness of an implementation

Our solutions:

- automatic generation of low-level descriptions (C and VHDL)
- include new arithmetic types and primitives in design tools (compilers, CAD tools)

Some Research Activities

Computer arithmetic for cryptography applications:

- modular arithmetic
- hyperelliptic curves
- implementation of basic crypto primitives
- **residue number systems**
- **double-base number systems**
- **addition chains for ECC implementations**
- secured arithmetic operators design
 - ▶ power-consumption aspects
 - ▶ fault injection
- implementations of applications
- **library**
- pairings

www.lirmm.fr/arith/

Residue Number Systems (RNS)

Contact: J.C. Bajard

- Public Key Cryptography uses **large** integers
- 192 to 521 bits for ECC, more than 1024 bits for RSA
- Residue Number Systems distribute large integer operations over small residues (parallel computations for + and ×)
- Operation with large integers made independently on each residue (non-positional number system)

Residue Number Systems (RNS)

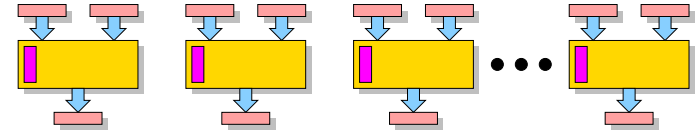
Theorem (Chinese Remainder Theorem)

(m_1, m_2, \dots, m_n) a set of coprime integers and $M = \prod_{i=1}^n m_i$. If (x_1, x_2, \dots, x_n) such that $x_i < m_i$, then there is a unique X such that:

$$0 \leq X < M \quad \text{and} \quad x_i = X \bmod m_i = |X|_{m_i} \quad \text{for } 1 \leq i \leq n$$

Definitions

- The set (m_1, m_2, \dots, m_n) of coprimes is called RNS basis.
- $X = (x_1, x_2, \dots, x_n)_{RNS}$, $0 \leq X < M$, $x_i = X \bmod m_i, \forall i$
- $A \pm B = (|a_1 \pm b_1|_{m_1}, \dots, |a_n \pm b_n|_{m_n})_{RNS}$
- $A \times B = (|a_1 \times b_1|_{m_1}, \dots, |a_n \times b_n|_{m_n})_{RNS}$



Residue Number Systems (RNS)

Example

Consider the RNS basis $\mathcal{B} = (3, 7, 13, 19)$.

All the numbers lower than $M = 5187$ are represented

$$\begin{aligned} X &= 147 & Y &= 31 \\ \bar{X} &= (0, 0, 4, 14)_{RNS} & \bar{Y} &= (1, 3, 5, 12)_{RNS} \end{aligned}$$

The addition and the multiplication mod M become:

$$\begin{aligned} \bar{X} + \bar{Y} &= (|0 + 1|_3, |0 + 3|_7, |4 + 5|_{13}, |14 + 12|_{19})_{RNS} \\ &= (1, 3, 9, 7)_{RNS} \\ &= 178 \\ \bar{X} \times \bar{Y} &= (|0 \times 1|_3, |0 \times 3|_7, |4 \times 5|_{13}, |14 \times 12|_{19})_{RNS} \\ &= (0, 0, 7, 16)_{RNS} \\ &= 4557 \end{aligned}$$

Residue Number Systems (RNS)

- [1, 2] Prime finite fields \mathbb{F}_p and modular arithmetic: RNS Montgomery multiplication algorithm, Leak Resistant arithmetic, Applications to RSA and ECC...
- [3] Arithmetic in \mathbb{F}_{2^k} using trinomial residue arithmetic
- [4] Lagrange representations \mathbb{F}_{p^k} of medium prime characteristic p

J.C. Bajard et L. Imbert, *A Full RNS Implementation of RSA*, IEEE Transactions on Computers, juin 2004 (Vol. 53, No. 6) p. 769-774

J.C. Bajard, L. Imbert, P.Y. Liardet, Y. Teglja, *Leak Resistant Arithmetic*, Workshop on Cryptographic Hardware and Embedded Systems CHES 2004, in LNCS, pages 62-75, Cambridge (Boston), USA, août 11-13, 2004.

J.C. Bajard, L. Imbert, and G. A. Jullien, *Parallel Montgomery Multiplication in $GF(2^k)$ using Trinomial Residue Arithmetic*, in Proceedings of the 17th IEEE symposium on Computer Arithmetic (ARITH 17) juin 2005, Cape Cod, MA, USA.

J.C. Bajard, L. Imbert et C. Nègre, *Arithmetic Operations in Finite Fields of Medium Prime Characteristic Using the Lagrange Representation*, IEEE Transactions on Computers, septembre 2006 (Vol. 55, No. 9) p. 1167-1177

Addition Chains

Contact: N. Meloni

Elliptic curve point scalar multiplication

Input: P a point of a curve E , an integer $k = \sum_{i=0}^{n-1} k_i 2^i$

Output: the point $Q = [k]P = \underbrace{P + P + P + \dots + P}_{k \text{ times}}$

Standard implementation: double-and-add

- 1: $Q \leftarrow P$
- 2: **for** i **from** $n-2$ **to** 0 **do**
- 3: $Q \leftarrow 2P$
- 4: **if** $k_i = 1$ **then** $Q \leftarrow Q + P$

Problem: **not resistant to SPA!**

Addition Chains

Some algorithms are “more” resistant to SPA: w -NAF recoding

In

$$k = \sum_{i=0}^{n-1} k_i 2^i, \quad k_i \in \{0, 1\}$$

process w digit at a time

$$|k_i| < 2^{w-1}$$

Example:

$$k = 267 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & \bar{1} & 0 & \bar{1} \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 3 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \bar{5} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 11 \end{pmatrix} \begin{matrix})_2 \\)_{2-NAF} \\)_{3-NAF} \\)_{4-NAF} \\)_{5-NAF} \end{matrix}$$

Leads to $n - 1$ DBL and $\frac{n}{w+1}$ ADD

Addition Chains

Addition chain (AC): finite sequence of integers v_1, \dots, v_n satisfying

$\forall l < n, v_l = v_i + v_j$ for some $i, j < n$

Euclidean addition chain: AC where $v_1 = 1, v_2 = 2, v_3 = v_1 + v_2$ and

$\forall 3 \leq i \leq n - 1$ if $v_i = v_{i-1} + v_j$ for some $j < i - 1$ then $v_{i+1} = v_i + v_{i-1}$

or $v_{i+1} = v_i + v_j$

advantage: resistant to SPA attacks

drawback: problem to find a short chain

Find a chain for the integer k : choose k' coprime with k and apply the subtractive form of Euclid's algorithm.

New formulae for the point addition:

input: P_1 and P_2 have the same Z

output: $ADD(P_1, P_2) = (P_1 + P_2, P_1)$ where $P_1 + P_2$ and P_1 have the same Z

Addition Chains

Example: $k = 34$, choose $k' = 19$

$$\begin{array}{rcl} 34 & - & 19 = 15 \\ 19 & - & 15 = 4 \\ 15 & - & 4 = 11 \\ 11 & - & 4 = 7 \\ 7 & - & 4 = 3 \\ 4 & - & 3 = 1 \\ 3 & - & 1 = 2 \\ 2 & - & 1 = 1 \\ 1 & - & 1 = 0 \end{array}$$

For $k = 34$, $k' = 21$ is better (Fibonacci)

Strategy: test several k' starting with $k' = k/\phi$

Current research: improvements, FPGA implementation for measurements (collab. UCC)

Double-Base Number Systems (DBNS)

Contact: L. Imbert

Redundant representation based on a sum of mixed powers of 2 and 3:

$$x = \sum_{i=1}^n x_i 2^{a_i} 3^{b_i}, \text{ with } x_i \in \{-1, 1\}, a_i, b_i \geq 0$$

Example: $127 = 108 + 16 + 3 = 72 + 54 + 1 = \dots$

	1	2	4	8	16
1					1
3	1				
9					
27			1		

	1	2	4	8
1	1			
3				
9				1
27		1		

Double-Base Number Systems (DBNS)

Smallest $x > 0$ requiring n terms in DBNS:

n	unsigned	signed
2	5	5
3	23	105
4	431	(4985)
5	18,431	?
6	3,448,733	
7	1,441,896,119	
8	?	

Theorem: Every positive integer x can be represented as a sum or difference of at most $O(\log x / \log \log x)$ terms

Example: 127 has exactly 783 DBNS representations, among which 6 are canonic: $127 = (108 + 18 + 1) = (108 + 16 + 3) = (96 + 27 + 4) = (72 + 54 + 1) = (64 + 54 + 9) = (64 + 36 + 27)$

Double-Base Number Systems (DBNS)

Application:

$$314159 = 2^4 3^9 + 2^8 3^1 - 1$$

$$[314159]P = [2^4 3^9]P + [2^8 3^1]P - P$$

cost: 12 DBL + 10 TPL + 2 ADD

$$314159 = 2^4 3^9 - 2^0 3^6 - 3^3 - 3^2 - 3 - 1$$

$$[314159]P = 3(3(3(3^3([2^4 3^3]P - P) - P) - P) - P) - P$$

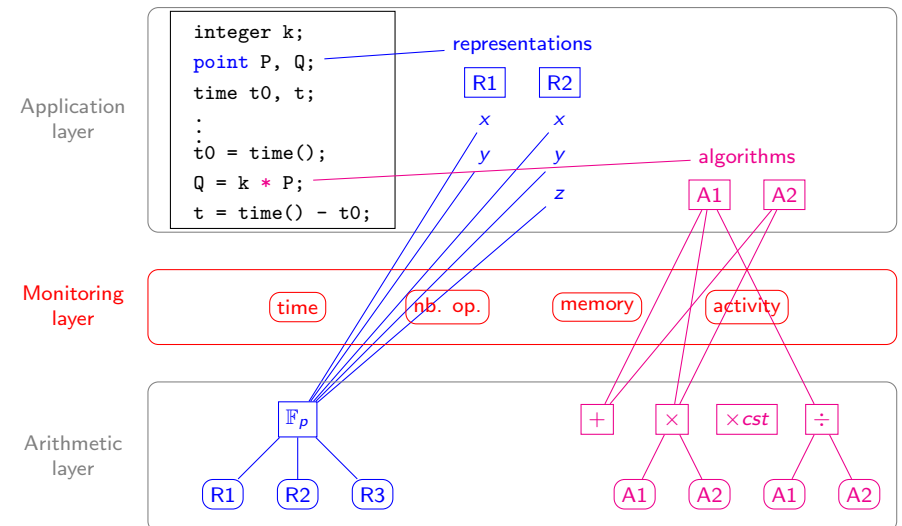
cost: 4 DBL + 9 TPL + 5 ADD

Goal: expansions with $a_1 \geq a_2 \geq \dots a_n \geq 0, b_1 \geq b_2 \geq \dots b_n \geq 0,$

$$x = \sum_{i=1}^n x_i 2^{a_i} 3^{b_i}, \text{ with } x_i \in \{-1, 1\}$$

We compute $[x]P$ in a Horner-like fashion (reuse partial results)

Library



Future Prospects

- improvements on basic arithmetic operators
 - ▶ speed
 - ▶ area
 - ▶ robustness against side-channel attacks
- pairings
- LLL for (NTRU signature, cryptanalysis)
- arithmetic operators robust to fault injection
- FPGA implementations of demonstrators
- library for prototyping the arithmetic level of crypto applications
- add new arithmetic types and operators into design tools
- arithmetic support for light crypto

The end, some questions ?

Contact:

- <mailto:arnaud.tisserand@lirmm.fr>
- <http://www.lirmm.fr/~tisseran>
- Arith group
- LIRMM Laboratory, CNRS–Univ. Montpellier 2
161 rue Ada. F-34392 Montpellier cedex 5. France

Thank you

Join us for ARITH18

<http://www.lirmm.fr/arith18>



18th IEEE Symposium on Computer Arithmetic
Montpellier, France, June 25-27, 2007

Submission deadline	October 15th, 2006
Acceptance notification	February 2007
Final version deadline	March 2007