# Useful Representation Systems for Cryptographic Implementations

## The French Connection

Jean Claude Bajard and Thomas Plantard

LIP6 CNRS-UPMC Sorbonne Universités
University of Wollongong

# Outline

# Residue Sytems

# Residue Sytems

# Residue Number System

Svoboda-Valach'57, Garner'59, Szabo-Tanaka'67, (CRT) Ch'in Chiu-Shao 1247

## RNS Base

- A set of coprime numbers $(m_1, ..., m_k)$, with $M = \prod_{i=1}^{k} m_i$

## Representation in RNS

- $A$ represented by its residues $(a_1, ..., a_k)$ with $a_i = |A|_{m_i}$

## Operations

- Full parallel operations $\pmod{M}$ with $M = \prod_{i=1}^{k} m_i$

  $(|a_1 \circ b_1|_{m_1}, \ldots, |a_n \circ b_n|_{m_n}) \to A \circ B \pmod{M}$

# Residue Number System: example

RNS Base:

$\mathcal{B} = (3, 7, 13, 19) \quad M = 5187$

Representations:

$$\begin{array}{llllll} X = 147 & & Y = 31 & & Z = 124 & \\ X_{RNS} = & (0,0,4,14) & Y_{RNS} = & (1,3,5,12) & Z_{RNS} = & (1,5,7,10) \end{array}$$

Operations:

$$\begin{array}{rlllll} X_{RNS} +_{RNS} Y_{RNS} & = (|0+1|_3, & |0+3|_7, & |4+5|_{13}, & |14+12|_{19}) \\ & = (1, & 3, & 9, & 7) \\ & = & 178 & & \\ X_{RNS} \times_{RNS} Y_{RNS} & = (|0\times 1|_3, & |0\times 3|_7, & |4\times 5|_{13}, & |14\times 12|_{19}) \\ & = (0, & 0, & 7, & 16) \\ & = & 4557 & & \end{array}$$

# Residue Sytems

# Lagrange representations in $GF(p^k)$ with $k \le p$

B .-Imbert-Negre $2006_{ieeeTC}$

## Extension of a finite field

Elements of $GF(p^k)$: $GF(p)$ polynomials of degree lower than $k$.

## Lagrange representation

- is defined by $k$ different points $e_1, ...e_k$ in $GF(p)$. ($k \le p$.)
- A polynomial $A(X) = \alpha_0 + \alpha_1 X + ... + \alpha_{k-1} X^{k-1}$ over $GF(p)$ is given in Lagrange representation by:

$$(a_1 = A(e_1), ..., a_k = A(e_k)).$$

- Remark: $a_i = A(e_i) = A(X) \bmod (X - e_i)$.

## Operations

are made independently on each $A(e_i)$ modulo $m_i(X)$
$m_i(X) = (X - e_i)$(as for FFT or Tom-Cook or Karatsuba).

# Example

## Finite Field

- $GF(23^5)$ defined by an irreducible polynomial $I := x^5 + 2\,x + 1$
- Let $A$ and $B$ be two elements of $GF(23^5)$ in polynomial forms: $A := 2\,x^4 + x + 3$ and $B := x^2 + 5\,x + 4$

## Lagrange representation

- We consider $GF(23^5)$ and the two sets of points: e = (2, 4, 6, 8, 10) and e' = (3, 5, 7, 9, 11).
- Then, elements are defined by:
  $A_e = (14, 13, 2, 15, 3)$ or $A_{e'} = (7, 16, 5, 1, 17)$
  $B_e = (18, 17, 1, 16, 16)$ or $B_{e'} = (5, 8, 19, 15, 19)$

# Trinomial residues in $GF(2^n)$

B.-Imbert-Jullien 2005$_{ARITH17}$

## Finite Field

Elements of $GF(2^n)$ are considered as $GF(2)$ polynomials of degree lower than $n$.

## Trinomial representation

- is defined by a set of $k$ coprime trinomials $m_i(X) = X^d + X^{t_i} + 1$, with $k \times d \geq n$,
- an element $A(X)$ is represented by $(a_1(X), ...a_k(X))$ with $a_i(X) = A(X) \bmod m_i(X)$.
- This representation is equivalent to RNS.

## Operations

are made independently on each $a_i(X)$ modulo $m_i(X)$

# Trinomial residues

## Example in $GF(2^n)$

We consider $d = 16$ and $k = 3$, thus $n \leq 48$:

- $base1 = (x^{16} + 1,\ x^{16} + x + 1,\ x^{16} + x^2 + 1)$
- $A := x^{18} + 1 \quad B := x^{23} + 1$
- $A_{base1} := (x^2 + 1,\ x^3 + x^2 + 1,\ x^4 + x^2 + 1)$
  $B_{base1} := (x^7 + 1,\ x^8 + x^7 + 1,\ x^9 + x^7 + 1)$

$AB_{base1} := (x^9 + x^2 + x^7 + 1,\ x^{11} + x^3 + x^9 + x^2 + x^8 + x^7 + 1,\ x^{13} + x^4 + x^2 + x^7 + 1)$

$A \times B := x^{41} + x^{23} + x^{18} + 1$

# Residue Systems

## Advantages

- Efficient Addition and Multiplication.
- Parallelization (GPU, multicore, ...).
- Small moduli.
- Side-Channel: Error Correction, Randomisation.

## Drawbacks

- $M$ smooth, not useful for Cryptography.
- Problems: modular reduction, euclidean division, comparison.
- Tool: Base conversion.

# Residue Sytems

# Residue version of Montgomery Reduction

Montgomery 1985, Posh and Posh 1995, B.-Didier-Kornerup 1997

## Residue Montgomery algorithm

1. $Q = -(Ap^{-1}) \bmod M$ (calculus in base $M$)
2. Extension of the representation of $Q$ to the base $M'$
3. $R = (A + Qp) \times M^{-1}$ (calculus in base $M'$)
4. Extension of the representation of $R$ to the base $M$

## Remarks

$R \equiv A \times M^{-1} \bmod p$ with $R < 2p$

Auxiliary bases $M'$, $M'$ and $M$ coprime (exact product, and existence of $M^{-1}$), $p < M, M'$ (or $\deg I(X) \leq \deg M(X), \deg M'(X)$)

## Montgomery notation

$A' = A \times M \bmod p$ and $\mathrm{Montg}(A' \times B', M, M', p) \equiv (A \times B) \times M \pmod p$

# Extension of Residue System Bases

- The extensions are similar to the polynomial interpolations.
- We consider $(a_1, ..., a_k)$ the residue representation of $A$ in base $M$.
- The Lagrange interpolation gives

$$\sum_{i=1}^{k} \left| a_i \times \left[ \frac{M}{m_i} \right]_{m_i}^{-1} \right|_{m_i} \times \frac{M}{m_i} = A + \alpha M$$

One has $\alpha = 0$ for polynomials. For integers $\alpha$ can be, according to the cases, neglected or computed.

# Extension in RNS Montgomery

B. - Didier - Kornerup 2001, Shenoy - Kumaresan 1989, Posh - Posh 1995, Kawamura - Koike - Sano - Shimbo 2000

- The extension of $Q$ from $M$ to $M'$ does not need to be exact, $Q$ is multiplied by $p$
- The second extension of $R$ from $M'$ to $M$ must be exact. Hence $\alpha$ must be determined
  - an extra modulo can be used

$$\alpha = \left| \left| \left| \sum_{i=1}^{k} \left| a_i \times \left[ \frac{M}{m_i} \right]_{m_i}^{-1} \right|_{m_i} \times \frac{M}{m_i} \right|_{m_{extra}} - a_{extra} \right|_{m_{extra}} \times M^{-1} \right|_{m_{extra}}$$

  - or we use the integer part of $\sum_{i=1}^{k} \left| a_i \times \left[ \frac{M}{m_i} \right]_{m_i}^{-1} \right|_{m_i} \times \frac{1}{m_i}$

# Exact Extension of Residue System Bases

## Newton interpolation, H.L. Garner 1958, B. - Kaihara - Plantard 2009

We first translate in an intermediate representation Mixed Radix Systems (MRS):

$$\begin{cases} \zeta_1 = a_1 \\ \zeta_2 = (a_2 - \zeta_1)\ m_1^{-1} \bmod m_2 \\ \zeta_3 = \Big((a_3 - \zeta_1)\ m_1^{-1} - \zeta_2\Big)\ m_2^{-1} \bmod m_3 \\ \quad \vdots \\ \zeta_n = \Big(\dots\Big((a_n - \zeta_1)\ m_1^{-1} - \zeta_2\Big)\ m_2^{-1} - \dots - \zeta_{n-1}\Big)\ m_{n-1}^{-1} \bmod m_n. \end{cases}$$

We evaluate $A$, with Horner's rule, as

$$A = (\dots((\zeta_n\, m_{n-1} + \zeta_{n-1})\ m_{n-2} + \dots + \zeta_3)\ m_2 + \zeta_2)\ m_1 + \zeta_1.$$

# Some conclusions about RNS

B. - Duquesne - Ercegovac - Meloni 2006, Szerwinski - Güneysu 2008, Guillermin 2010, Antão - B. - Sousa 2010

- RNS is well adapted to parallel architectures (GPU, Multicore,...).
- Modular reductions stay costly.
- For ECC or Pairing it is possible to reduce the number of modular reductions since $A \times B + C \times D$ needs only one reduction.
- As for the interpolation, the choice of the bases is important. Does there exist an FFT like approach for RNS?

# Modular Positional Arithmetics

# Modular Positional Arithmetics

# Positional Number Systems and Modular Operations

- Number system: radix $\beta$ and a set of digits $\{0, ..., \beta - 1\}$.
- We denote by $p$ the modulo, with $p < \beta^n$

$$\beta^n \equiv \varepsilon \pmod{p}, \text{ with } \varepsilon = \sum_{i=0}^{n-1} \varepsilon_i \beta^i,\ \varepsilon_i \in \{0, ..., \beta - 1\}$$

- A modular operation (ex.: modular multiplication)
  1. Polynomial operation: $W(X) = A(X) \times B(X)$
  2. Polynomial reduction: $V(X) = W(X) \bmod (X^n - \varepsilon(X))$
     - Pseudo-Mersenne properties for the reduction.
     - The coefficients of $V(X)$ can be larger than $\beta - 1$ the maximal digit.
  3. Coefficient reduction: $M(X) = \text{Reductcoeff}(V(X))$

# Modular Reduction with pseudo-Mersenne numbers

$p = \beta^n - \varepsilon$ avec $0 \leq \varepsilon < \beta^{n/2}$

- In this kind of reduction we have two products by $\varepsilon$
  - $\varepsilon$ very small, for example $\varepsilon < \beta$, for having a product by a digit
  - $\varepsilon$ very sparse (most of the digits are equal to zero) then the product is replaced by some shift-and-adds.
- There are only very few such Pseudo-Mersenne numbers.
- The question is: Is it possible to have a number system where $p$ is a Pseudo-Mersenne number?

# Modular Arithmetic Adapted Bases

Th. Plantard PhD 2005

## The main idea

- Representation of $A$:
  $$A = \sum_{i=0}^{n-1} a_i \gamma^i \bmod p, \text{ with } a_i \in \{0, ..., \rho - 1\} \text{ and } p < \rho^n.$$
- $\gamma$ can be huge, but $\rho$ is small (redundancy).
- $(p, n, \gamma, \rho)$ defines the MAAB system.

## Modular reduction

- For the polynomial reduction: $\gamma^n \equiv \varepsilon \pmod{p}$ with $\varepsilon$ small.
- For the coefficient reduction different approaches.

# Modular Arithmetic Adapted Bases

B. - Imbert - Plantard $2004_{SAC}$

## First approach (find $P$ and $\gamma$)

- The construction of the system giving some features: $n = 8$, and $\rho = 2^{32}$ with $p < \rho^8$ determine the size of the problem.
- The property $\gamma^8 \equiv 2 \pmod{p}$ for the polynomial reduction.
- The coefficient reduction is given by $2^{32} \equiv \gamma^5 + 1 \pmod{p}$

Thus $V = 2^{32}V_1 + V_0 = 2^{32}Id.V_1 + V_0 \equiv M.V_1 + V_0 \pmod{p}$ with

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 1 \end{pmatrix} \equiv \begin{pmatrix} 2^{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2^{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2^{32} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2^{32} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2^{32} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2^{32} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2^{32} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2^{32} \end{pmatrix} \pmod{p}$$

# Modular Arithmetic Adapted Bases

B. - Imbert - Plantard $2004_{SAC}$

## Remarks and construction

- $2^{32}Id - M = 0 \bmod p$ defines a lattice.
- $p$ divides $\det(2^{32}Id - M)$, a factorization gives:

  $p = 115792089021636622262124715160334756877804245386980633020041035952359812890593$

  which corresponds to the expected size.
- The value of $\gamma$ is deduced as a solution of $\gcd(X^8 - 2, 2^{32} - X^5 - 1)$ modulo $p$:

  $\gamma = 14474011127704577782765589395224532314179217058921488395049827733759590399996$
- Generally, $M$ is found with coefficients lower than $2^{k/2}$, which means that three rounds are sufficient.

# Modular Arithmetic Adapted Bases

B. - Imbert - Plantard $2005_{ARITH}$

## Second approach (find $\rho$ and $\gamma$)

Consider the modulo $p = 53$, and $n = 7$ for the digit size, $p < \rho^7$, and we expect a small value for $\rho$ like $\rho = 2$.
We look for a radix with Pseudo-Mersenne property, we find $\gamma = 14$, such that $\gamma^7 \equiv 2 \pmod{p}$.
We consider the carry propagation lattice modulo $p$

$$L = \begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{pmatrix} = \begin{pmatrix} -14 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -14 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -14 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -14 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -14 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -14 & 1 \\ 53 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# Modular Arithmetic Adapted Bases

B. - Imbert - Plantard $2005_{ARITH}$

## Remarks and construction

- This lattice $L$ admits as short vector
  $(1,1,0,0,0,0,1) = V_6 + 14 * V_5 + 14^2 * V_4 + 14^3 * V_3 + 14^4 * V_2 + (14^5+1) * V_1 + 142067.V_7$.
- With $\gamma^7 \equiv 2 \pmod{p}$, we construct a sublattice $L'$.

$$\Rightarrow L' = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 & 2 & 1 \end{pmatrix}$$

- Hence, $\rho$ can be chosen equal to 2.
- Coefficient reduction becomes a closest vector problem.

# Modular Arithmetic Adapted Bases

## Conclusions

- First approach: efficient coefficient reduction but reduced choice of moduli.
- Second approach: we can choose the moduli but complexity of the coefficient reduction.

# Modular Positional Arithmetics

# Ostrowski Bases

## Continued Fraction Expansion of $\frac{a}{m}$

- $\frac{a}{m} = k_0 + \cfrac{1}{k_1 + \cfrac{1}{k_2 + \cfrac{1}{k_3 + \dots}}}$ et $\frac{p_i}{q_i} = k_0 + \cfrac{1}{k_1 + \cfrac{1}{k_2 + \dots \frac{1}{k_i}}}$
- $\theta_i = aq_i - mp_i$
- Recursive computation

$$\begin{array}{llll} q_{i+2} & = k_{i+2}q_{i+1} + q_i & q_0 = 1 & q_{-1} = 0 \\ \theta_{i+2} & = k_{i+2}\theta_{i+1} + \theta_i & \theta_0 = a - mk_0 & \theta_{-1} = -m \end{array}$$

## Ostrowski representations base $(q_i)$ and base $(\theta_i)$

$$b = \sum_{i=0}^{n-1} b_i q_i, \quad \text{with } b_0 < k_1,\ 0 \le b_i \le k_{i+1},\ b_i = 0 \ \text{ if } \ b_{i+1} = k_{i+2}$$

$$x = \sum_{i=0}^{n-1} x_i \theta_i, \quad \text{with } x_0 < k_1,\ 0 \le x_i \le k_{i+1},\ x_i = 0 \ \text{ if } \ x_{i+1} = k_{i+2}$$

# Ostrowski Bases

## Example

### Continued Fraction Expansion of $\frac{3238}{7741}$

- $\frac{3238}{7741} = [0; 2, 2, 1, 1, 3, 1, 2, 4, 1, 2, 3]$
- Ostrowski base $(q)$

$$B_q := [1, 2, 5, 7, 12, 43, 55, 153, 667, 820, 2307]$$

- Consider $b = 6000$ in Ostrowski representation

$$b_{B_q} := [0, 1, 0, 1, 0, 1, 1, 3, 0, 1, 2]$$

- $x := [1, 0, 1, 0, 3, 0, 2, 0, 1, 0, 3]$ represents 7740 the largest value

# Ostrowski Bases
## Example

### Continued Fraction Expansion of $\frac{3238}{7741}$

- $\theta$ base

$$B_\theta := [3238, -1265, 708, -557, 151, -104, 47, -10, 7, -3, 1]$$

- Decreases and Alternates
- $x := [1, 0, 1, 0, 3, 0, 2, 0, 1, 0, 3]$ represents 4503 the largest value
- $y := [0, 2, 0, 1, 0, 1, 0, 4, 0, 2, 0]$ represents $-3237$ the smallest value
- Remark: $x - y = 7740$

# Ostrowski Bases and Multiplication

M. Gouicem PhD 2013

## Computation of $a \times b \mod m$

1. Evaluation of $q_i$ and $\theta_i$ from $\frac{a}{m}$.
2. Representation of $b$ in the Ostrowski base $(q_i)$.

$$b = \sum_{i=0}^{n-1} b_i q_i, \quad \text{with } b_0 < k_1,\ 0 \leq b_i \leq k_{i+1},\ b_i = 0 \ \text{ if } \ b_{i+1} = k_{i+2}$$

3. Return $R = \sum_{i=0}^{n-1} b_i \theta_i = a \cdot b \mod m$, with $(-m < R < m)$

Proof: $\sum_{i=0}^{n-1} b_i \theta_i = \sum_{i=0}^{n-1} b_i (a q_i - m p_i) = a \sum_{i=0}^{n-1} b_i q_i + \alpha m$

# Ostrowski Bases

## Example

### Multiplication of $3238 \times 6000 \pmod{7741}$

- $\frac{3238}{7741} = (0, 2, 2, 1, 1, 3, 1, 2, 4, 1, 2, 3)$

  $B_q := [1, 2, 5, 7, 12, 43, 55, 153, 667, 820, 2307]$

  $B_\theta := [3238, -1265, 708, -557, 151, -104, 47, -10, 7, -3, 1]$

- Consider $b = 6000$ in Ostrowski representation

  $b_{B_q} := [0, 1, 0, 1, 0, 1, 1, 3, 0, 1, 2]$

- We obtain in $\theta$ base

  $(1*(-1265)+1*(-557)+1*(-104)+1*47+3*(-10)+1*(-3)+2*1)$

  $= (-1910) \equiv 5831 \equiv 3238 \times 6000 \bmod 7741$

## Conclusions

- Quadratic complexity in the size of the modulo.
- Division: the $\theta$ representation provides the division in Ostrowski representation.
- Allow to perform inversion, multiplication and division with the same circuit.
- Multiplications and/or divisions by the same number $a$ becomes efficient

# Exponent representations (ECC $kP$)

# Exponent representations (ECC $kP$)

# Addition Chains: Fibonacci - Zeckendorf

## Representation of Zeckendorf - 1972 (1939)

- Fibonacci Series: $F_{n+2} = F_{n+1} + F_n$, with $F_0 = 0$ and $F_1 = 1$
  $1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots$
- Representation with $q_i = F_{i+2}$

$$b = \sum_{i=1}^{n-1} b_i q_i, \quad \text{with } b_i \in \{0, 1\},\ b_i = 0 \text{ if } b_{i+1} = 1$$

## Remarks

- It is the Ostrowski representation using the continued fraction expansion of the golden ratio.
- Example: $k := 1117 = [0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1]_{\mathcal{Z}} = F_3 + F_5 + F_9 + F_{11} + F_{16} = 2 + 5 + 34 + 89 + 987$

# Addition Chains: Fibonacci - Zeckendorf

*kP* with an efficient $P + Q$.

- Algorithm:
  1. Decomposition in the Fibonacci representation
  2. Recursive computing with respect to the decomposition
- Example: Evaluation right to left of $1117.P$ using $[0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1]_{\mathcal{Z}}$ with 18 Additions

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | |
| 1 | 1 | 2 | 3 | 5 | 8 | | | | | | | | | | |
| | | | | | 9 | 14 | 23 | | | | | | | | |
| | | | | | | | 24 | 38 | 62 | 100 | 162 | | | | |
| | | | | | | | | | | | 163 | 263 | 426 | | |
| | | | | | | | | | | | | | 427 | 690 | 1117 |

# Addition Chains: Fibonacci - Zeckendorf

E. B. Burger et al. $2012_{ActaAr}$.

## Properties

- Length: $k$ such that $F_k \leq n < F_{k+1}$
- Ratio of ones: $\frac{\psi(k)}{k} \rightarrow \frac{5-\sqrt{5}}{10} = 0.2763$

## Pros and cons

- Advantage: only additions
- Drawback: more digits than in binary: ratio $= \frac{\ln 2}{\ln \varphi} \sim 1.44$ with $\varphi = \frac{1+\sqrt{(5)}}{2}$
- Tool: Greedy Algorithm

# Euclidean Addition Chains

N. Meloni PhD 2007, Herbaut-Liardet-Meloni-Teglia-Veron $2010_{INDOCRYPT}$

## Definition

A Euclidean addition chain (EAC) of length s for an integer $k$ is a sequence $(c_i)_{i=1\ldots s}$ with $c_i \in \{0,1\}$.
The computation of $k$ is obtained from the sequence $(v_i, u_i)_{i=0..s}$
$v_0 = 1,\ u_0 = 2$
$(u_i, v_i) = (v_{i-1} + u_{i-1}, v_{i-1})$ if $c_i = 1$ (small step),
$(u_i, v_i) = (v_{i-1} + u_{i-1}, u_{i-1})$ if $c_i = 0$ (big step).
Then we denote $\chi(c) = v_s + u_s = k$.

## Properties

- Euclidean algorithm scheme
- $\chi(0_n) = F_{n+4},\ \chi(1_n) = n + 3$

# Euclidean Addition Chains

N. Meloni PhD 2007, Herbaut-Liardet-Meloni-Teglia-Veron 2010$_{INDOCRYPT}$

### Example

We can find shortest chains for 1117 with 15 additions:

$$[1117, 648], [648, 469], [469, 179], [290, 179], [179, 111], [111, 68], [68, 43], [43, 25], [25, 18], [18, 7], [11, 7], [7, 4], [4, 3], [3, 1], [2, 1], [1, 1]$$

$\chi(01000100000010) = 1117$

### Construction of keys

How to construct a set of keys with efficient EAC representations?

# Exponent representations (ECC $kP$)

# Double base

Dimitrov-Jullien-Miller 1999$_{ieeeTC}$, Dimitrov-Imbert-Mishra 2005$_{ASIACRYPT}$

## Double Base

- Representation: $X = \sum x_{i,j} 2^i 3^j, \quad x_{i,j} \in \{0,1\}$
- Example: $127 = 1111111_b = 2^3 3^2 + 2^1 3^3 + 2^0 3^0 = 72 + 54 + 1$

## $kP$ with $2P$ and $3P$

1. Decomposition in double base, find a path.
2. Return $2^{i_0} 3^{j_0} P + 2^{i_1} 3^{j_1} P + 2^{i_2} 3^{j_2} P + \dots$

## Advantages and Drawbacks

- Sparse representation
- Redundancy and optimal representation

# Double base

Berthé - Imbert 2009$_{DMTCS}$, Tijdeman 1974$_{CompMath}$

## Construction

- How to find the nearest $2^a 3^b$ to a given number $N$?
- Then a greedy algorithm can be used.
- Number of non-zero digits is in $O(\log N / \log\log N)$

## Method

- We minimize: $a * \ln 2 + b * \ln 3 - \ln N$ or $a \log_3 2 + b - \log_3 N$
- Considering the fractional part we have $(a \log_3 2 - \log_3 N) \bmod 1$

# Double base

Berthé - Imbert 2009$_{DMTCS}$

## Method using Ostrowski

- We consider the continued fraction expansion of $\log_3 2$ $[0; 1, 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, \ldots]$
- The Ostrowski bases are constructed
  - $\theta_i = q_i * \log_3 2 - p_i$
  - Recursive computation

$$\begin{array}{llll} q_{i+2} & = k_{i+2} q_{i+1} + q_i & q_0 = 1 & q_{-1} = 0 \\ \theta_{i+2} & = k_{i+2} \theta_{i+1} + \theta_i & \theta_0 = \log_3 2 - k_0 & \theta_{-1} = -1 \end{array}$$

- $a$ is found in two steps
  - Representation of $\log_3 N \bmod 1$ in $\theta$ base:

$$(\log_3 N) \bmod 1 = \sum_{i=0}^{n-1} n_i \theta_i$$

  - We have $a = \sum_{i=0}^{n-1} n_i q_i$

# Double base

Berthé - Imbert 2009$_{DMTCS}$

## Example for $N = 2000$

- We consider the continued fraction expansion of $\log_3 2$: $[0; 1, 1, 1, 2, 2]$
  and the bases: $B_q = [1, 1, 2, 3, 8, 19]$
  $B_\theta = [0.63, -0.369, 0.26, -.1, 0.047, -0.012]$
- we consider $T = (\log_3 N - \lfloor \log_3 N \rfloor) = 0.918639575$
  - $T_\theta = [1, 0, 1, 0, 0, 0] = 0.8927892604$
  - In the base $B_q$: $[0, 0, 1, 0, 0, 0] = 3 = a$
  - Then $\lfloor \log_3(N/2^3) \rfloor = 5 = b$
- We verify that:

| | | | | | | |
|---|---|---|---|---|---|---|
| $2^1 3^6$ | $2^3 3^5$ | $2^4 3^4$ | $2^6 3^3$ | $2^7 3^2$ | $2^9 3^1$ | $2^{10} 3^0$ |
| 1458 | 1944 | 1296 | 1728 | 1152 | 1536 | 1024 |

# Conclusions

Residue Sytems
- Residue Number System
- Polynomial Residue Representations
- Modular Reduction

Modular Positional Arithmetics
- Modular Arithmetic Adapted Bases
- Ostrowski Bases

Exponent representations (ECC $kP$)
- Addition Chains
- Double base

Conclusions

# Tools and open problems

## Residue Systems

- Chinese Remainder Theorem, Polynomial interpolations
- Find good bases (base extension)

## Modular Positional representations

- Lattice reduction, Shortest vector, Closest vector
- Continued Fraction Expansion, Ostrowski representation

## Exponent representation

- Fibonacci series, Zeckendorf, Euclid algorithm
- Shortest addition chains, Ostrowski approximation