Number systems and Cryptography

JC Bajard

LIP6, UPMC
4 place Jussieu, 75005 Paris, France

Hue August 2012

Contents

Arithmetic for Cryptography

Key Agreement

Diffie-Hellman (1976)

Construction p a prime number and g a generator of $\mathbb{Z}_p^*$ [1]

Character A selects a random x, and sends $a = g^x \bmod p$ to B

Character B selects a random y, and sends $b = g^y \bmod p$ to A

Common Secret A constructs $k = b^x \bmod p$ and B constructs $k' = a^y \bmod p$, thus $k = k'$

[1] in fact, the order of g equals to the biggest factor of $p-1$ is sufficient

Public Key Cryptosystem

- **RSA** (1978) Robustness due to factorization
  Construction $n = p * q$, p and q two large primes,
  e, and d such that $e \times d \equiv 1 \pmod{\phi(n)}$.
  Sender message m, $m < n$, computes $c = m^e \bmod n$ and sends c
  Receptor Secret key d, computes $m = c^d \bmod n$
- **El Gamal** (1985) Based on DH, with (p, g, g^a) as public key, Robustness due to Discrete Logarithm Problem
- **ECC** Koblitz-Miller (1985) Law group of the points of an elliptic curve curve defined on a finite field $\mathbb{F}_p$. DH or El Gamal can be applied using k times a point P (generator).

Arithmetic point of view

- Multiplication with huge numbers (300 to 2000 bits)
- Modular reduction, without division (division is costly)
- Exponentiation, using the representation of the exponent
- Exponent most of the time is secret

Evaluation of the product $P = A \times B$

1. Considering numbers as polynomials
$$A = \sum_{i=0}^{k-1} a_i \beta^i \rightarrow A(X) = \sum_{i=0}^{k-1} a_i X^i$$
2. Evaluation of a polynomial product: $P(X) = A(X) \times B(X)$
   - using a matrix-vector product
   - using interpolation: $i = 0 \ldots k, \quad A(e_i) = \sum_{i=0}^{k-1} a_i e_i^i$
3. Evaluation of the value: $P(\beta) = A(\beta) \times B(\beta)$

Evaluation of the product $P = A \times B$

- Karatsuba-Ofman (1962): with $A(X) = A_1X + A_0$ and $e_0 = 0$, $e_1 = -1$ et $e_2 = \infty$, complexity $K(n) = O(n^{\log_2(3)})$
- Toom-Cook (1963-1966): with $A(X) = A_2X^2 + A_1X + A_0$ and $e_0 = 0$, $e_1 = -1$, $e_2 = 1$, $e_3 = 2$ et $e_4 = \infty$, complexity $T_3(n) = O(n^{\log_3(5)})$... $T_k(n) = O(n^{\log_k(2k-1)})$.
- Schönhage-Strassen (1971): ω primitive n^{th} root of unity, $\omega^n = 1$, $e_i = \omega^i$ for $i = 0..n-1$, complexity $FFT(n) = O(n \log n \log\log n)$.
- But in practice the school-book algorithm in $O(n^2)$ is sufficient[2]

[2] see gmp-mparam.h of GMP for the different architectures x_86-64 Karatsuba 28 (1792), Toom-Cook 81 (5184), FFT 7552

Montgomery Reduction (1985)[3]

Montgomery(A P)

Input $\beta^{n-1} \leq P < \beta^n$ et $A < P\beta^n < \beta^{2n}$

Output , $R = A \times \beta^{-n} \bmod P$

Core $Q \leftarrow A \times |P^{-1}|_{\beta^n} \bmod \beta^n$

$R \leftarrow (A - Q \times P) \div \beta^n$, $(R < 2P)$

While $R \geq P$ do $R \leftarrow R - P$

Complexity : 2 products of n digits (close to 2 half products)

- Montgomery notation: $\widetilde{A} = A \times \beta^n \bmod P$
- $\widetilde{A} = \text{Montgomery}(A \times |\beta^{2n}|_P, P)$
- $\widetilde{A} + \widetilde{B} = \widetilde{A + B}$ et $\widetilde{AB} = \text{Montgomery}(\widetilde{A} \times \widetilde{B}, P)$

[3] 1986 Barrett algorithm 40

Residue Number Systems

Definition of the Residue Number Systems

- Issue from the Chinese Remainder Theorem[4] , introduced in computer arithmetic in 1957-1967 [5].
- Residue Number System
  - RNS base: a set of coprime numbers $(m_1, \dots, m_k)$
  - RNS representation: $(a_1, \dots, a_k)$ with $a_i = |A|_{m_i}$
  - Full parallel operations $\bmod M$ with $M = \prod_{i=1}^{k} m_i$
    $(|a_1 \otimes b_1|_{m_1}, \dots, |a_n \otimes b_n|_{m_n}) \rightarrow A \otimes B \pmod{M}$
- Very fast addition and product, but comparison and division are costly.

[4] Ch'in Chiu-Shao 1247

[5] 1957 Svoboda and Valach, 1959 Garner, 1967 Szabo and Tanaka

Residue Number Systems: example

$$\text{Modular system: } \mathcal{B}_m^4 = \{3, 7, 13, 19\} \quad M = 5187$$

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

$$\begin{array}{lllllll} X_{RNS} +_{RNS} Y_{RNS} & = \{ & |0+1|_3, & |0+3|_7, & |4+5|_{13}, & |14+12|_{19} & \} \\ & = \{ & 1, & 3, & 9, & 7 & \} \\ & = & 178 & & & & \end{array}$$

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

Remarks about the complexities of RNS

- We consider m the biggest element of the RNS base
- $\Phi(m) = \sum_{\substack{p \leq m \\ p \text{ prime}}} \log p = \log \prod_{\substack{p \leq m \\ p \text{ prime}}} p \sim m$
- If $2^{m-1} \leq M < 2^m$ then the size moduli is of order $\mathcal{O}(\log m)$.
- In other words, if addition and multiplication have complexities of order $\Theta(f(m))$ then in RNS the complexities become $\Theta(f(\log m))$.

Residue version of Montgomery Reduction[6]

- The residue base is such that $p < M$
- We use an auxiliary base such that $p < M'$, M' and M coprime.
- Steps of the 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
- The values are represented in the two bases.

[6] Posh and Posh 1995, B.-Didier-Kornerup 1997

Extension of Residue System Bases (from M to M')

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

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

The factor α can be, in certain cases, neglected or computed.

- Another approach consists in the Newton interpolation where A is correctly reconstructed.

Extension in RNS Montgomery

- The extension of Q from M to M' does not need to be exact, Q is multiply by p (Annex 41) [7]
- The second extension of R from M' to M must be exact. Hence α must be determined,
  - an extra modulo can be used (Annex 6) [8]
  - or from the interger part of $\sum_{i=1}^{k} \left| a_i \times \left[\frac{M}{m_i}\right]^{-1}_{m_i} \right|_{m_i} \times \frac{1}{m_i}$ [9]

[7] B. - Didier -Kornerup 2001
[8] Shenoy - Kumaresan 1989
[9] Posh - Posh 1995, Kawamura - Koike - Sano - Shimbo 2000

Exact Extension of Residue System Bases (Newton interpolation)

We first translate in an intermediate representation Mixed Radix Systems (MRS):[10]

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

We evaluate A, with Horner's rule, as

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

[10] H.L. Garner 1958

Some conclusions about RNS

- 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 based on the fact the $A \times B + C \times D$ needs only one reduction.
- As for interpolation, the choice of the bases are important. Does it exist a FFT like approach for RNS ?

Lattices and Modular Reduction

Pseudo Mersenne and Reduction [11]

When possible p can be chosen to facilitate the reduction
$p = \beta^n - \xi$ with $0 \leq \xi < \beta^{n/2}$ $(\xi^2 \leq \beta^n - 2\beta^{n/2} + 1)$.

For reducing C (e.g. $C = A \times B \leq (p-1)^2$), we note
$C = C_1\beta^n + C_0$

- First step of reduction: $C \equiv (C' = C_1\xi + C_0) \pmod{p}$
  $C' = C_1'\beta^n + C_0'$ with $C_1' \leq \xi$ and $C_0' \leq \beta^n - 1$
- Second step of reduction: $C' \equiv (C'' = C_1'\xi + C_0') \pmod{p}$
  with $C'' + \xi < \beta^n + p$
- Final step: If $C'' + \xi \geq \beta^n$ Then $R = C'' + \xi - \beta^n$ else $R = C''$

[11] Crandall 1992

Réduction modulaire

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

- In this kind of reduction we have two products by ξ,
  - ξ very small, for example $\xi < \beta$, for having a product by a digit
  - ξ very sparse (most of the digit are equal to zero) then the product is replaced by some shift-and-adds.
- Such Pseudo-Mersenne numbers are very few. Furthermore for different reasons it could be not possible to have a pseudo-Mersenne (i.e. RSA $N = pq$)
- The question is, Is it possible to have a number system where p have this kind of properties??

Lattices and Modular Systems

- Number system: radix β and a set of digits $\{0, ..., \beta - 1\}$.
- We denote by p the modulo, with $p < \beta^n$,
$$\beta^n \equiv \sum_{i=0}^{n-1} \epsilon_i \beta^i \pmod{p} \text{ with } \epsilon_i \in \{0, ..., \beta - 1\}$$
- A modular operation (for example: a modular multiplication):
  1. Polynomial operation: $W(X) = A(X) \bigotimes B(X)$
  2. Polynomial reduction : $V(X) = W(X) \bmod (X^n - \sum_{i=0}^{n-1} \epsilon_i X^i)$
     - Pseudo-Mersenne properties for the reduction.
     - The coefficients of $V(X)$ can be bigger than $\beta - 1$ the maximal digit.
  3. Coefficient reduction :$M(X) = \text{Reductcoeff}(V(X))$

Lattices and Modular Systems

Lattice approach

- A number $A = \sum_{i=0}^{n-1} a_i \beta^i$ corresponds to a vector $(a_0, \dots, a_{n-1})$
- We consider the lattice defined by the representations of zero modulo p, equivalent to a combination of the carry propagation and the modular reduction:

$$\begin{pmatrix} -\beta & 1 & \dots & 0 & 0 \\ 0 & -\beta & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -\beta & 1 \\ p & 0 & \dots & 0 & 0 \end{pmatrix} \begin{array}{c} \leftarrow \text{lattice} \\ (\det = p) \\ \text{sublattice} \rightarrow \\ (\det = \beta^n - \epsilon) \end{array} \begin{pmatrix} -\beta & 1 & \dots & 0 & 0 \\ 0 & -\beta & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -\beta & 1 \\ \epsilon_0 & \epsilon_1 & \dots & \epsilon_{n-2} & (\epsilon_{n-1} - \beta) \end{pmatrix}$$

- The goal is to find a vector G of the lattice (or sublattice) such that $V - G$ has all its coefficients equal to digits (close vector).

Lattices and Modular Systems

Example

For $P = 97$ and $\beta = 10$, we have $10^2 \equiv 3 \pmod{P}$. We consider the lattice:

$$\begin{pmatrix} B_0 \\ B_1 \end{pmatrix} = \begin{pmatrix} -10 & 1 \\ 3 & -10 \end{pmatrix}$$

Let $V(25, 12) = 25 + 12\beta$.

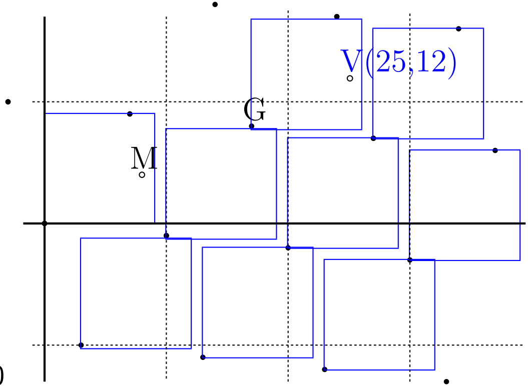
For reducing V, we determine $G(17, 8) = -2B_0 - B_1$ a vector of the lattice close to V.

Thus , $V(25, 12) \equiv M(8, 4) = V(25, 12) - G(17, 8)$.
We verify that $25 + 120 = 145 \equiv 48 \pmod{97}$

Lattices and Modular Number Systems

Example

The reduction is equivalent to find a close vector.
Let $G(X)$ be this vector, then $M(X) = V(x) - G(X)$



$P = 97$ $\beta = 10$

Lattices and Modular Systems

A new system

- Polynomial reduction depends of the representation of $\beta^n \pmod{P}$
- In Thomas Plantard's PhD (2005),β can be as large as P : $\beta^n \equiv \epsilon \pmod{P}$, for obtaining a set of digits $\{0, ..., \rho - 1\}$ where ρ is small

Example: Let us consider a MNS defined with $P = 17, n = 3, \beta = 7, \rho = 3$. Over this system, we represent the elements of $\mathbb{Z}_{17}$ as polynomials in β, of degree at most 2, with coefficients in $\{0, 1, 2\}$

Lattices and Modular Systems

A new system

0	1	2	3	4	5
0	1	2	$\beta + 2\beta^2$	$1 + \beta + 2\beta^2$	$2 + \beta + 2\beta^2$
6	7	8	9	10	11
$1 + \beta + \beta^2$	β	$1 + \beta$	$2 + \beta$	$2\beta + 2\beta^2$	$1 + 2\beta + 2\beta^2$
12	13	14	15	16	
$2\beta + \beta^2$	$1 + 2\beta + \beta^2$	2β	β^2	$1 + \beta^2$	$2 + \beta^2$

The system is clearly redundant.
For example: $5 = \beta + \beta^2 = 2 + \beta + 2\beta^2$, or
$14 = 2\beta = 2 + 2\beta + \beta^2 = 1 + 2\beta^2$.

Lattices and Modular Systems

Construction of Plantard Systems

- In a first approach, n and $\rho = 2^k$ are fixed. The lattice is constructed from the representation of ρ in the number system. P and β are deduced. Efficient algorithm for finding a close vector. $\boxed{48}$
- In a general approach, where P, β and n are given, the determination of ρ is obtained by reducing with LLL (Lenstra Lenstra Lovasz, 1982). No efficient algorithm for finding a close vector. $\boxed{46}$

Signed Digit Number Systems

Redundant Number Systems: Avizienis (1961) [12]

- Redundant Number Systems
  Signed digits: $x_i \in \{-a, \ldots, -1, 0, 1, \ldots, a\}$ Radix β with $a \leq \beta - 1$.
- Properties
  - If $2a+1 \geq \beta$, then each integer has at least one representation. An integer X, with $-a\frac{\beta^n-1}{\beta-1} \leq X < a\frac{\beta^n-1}{\beta-1}$, admits a (unique if $=$) representation

$$X = \sum_{i=0}^{n-1} x_i \beta^i \quad \text{with } x_i \in \{-a, \cdots -1, 0, 1, \ldots, a\}$$

  - If $2a \geq \beta + 1$, then we have a carry free algorithm. $\boxed{43}$
- Borrow-save (Duprat, Muller 1989): extension to radix 2.

[12] Cauchy 1840

Example: radix 10, $a = 9$ $\boxed{43}$

$$
\begin{array}{rl}
\bar{2}3594\!\!\!\bar{}\,2 & (= -164138) \\
+\ 461\bar{6}7 & (= 46047) \\
\hline
001\bar{1}10 & (= t) \\
\bar{2}7100\bar{1} & (= w) \\
\hline
\bar{2}82\bar{1}1\bar{1} & (= s = -118091)
\end{array}
$$

Properties of the signed digits redundant systems

- **Advantages:**
  - Constant time carry-propagation-free addition
  - Large radix: parallelisation
  - Small radix: fast circuits, on-line calculus
  - Increasing of the performances of the algorithms based on the addition
- **Drawbacks:** comparisons, sign...

Non-Adjacent Form

- This representation is inspired from **Booth recoding (1951)** used in multipliers.
- **Definition of NAF_w recoding:** (Reitwiesner 1960) Let k be an integer and $w \geq 2$. The non-adjacent form of weight w of k is given by $k = \sum_{i=0}^{l-1} k_i 2^i$ where $|k_i| < 2^{w-1}$, $k_{l-1} \neq 0$ and each w-bit word contains at most one non-zero digit.

1. For a given k, $NAF_w(k)$ is unique.
2. For a given $w \geq 2$, the length of $NAF_w(k)$ is at most equal to the length of k plus one.
3. The average density of non-zero digits is $1/(w+1)$.

NAF_w Examples

We consider $k = 31415592$.

$$
\begin{array}{rcccccccc}
k_2 = & 1 & 1\,1\,0\,1 & 1\,1\,1\,1 & 0\,1\,0\,1 & 1\,1\,0\,1 & 0\,0\,1\,0 & 1\,0\,0\,0 \\
NAF_2(k) = & 1\,0 & 0\,0\,\bar{1}\,0 & 0\,0\,0\,0 & \bar{1}\,0\,\bar{1}\,0 & 0\,\bar{1}\,0\,1 & 0\,0\,1\,0 & 1\,0\,0\,0 \\
NAF_3(k) = & 1\,0 & 0\,0\,\bar{1}\,0 & 0\,0\,0\,\bar{1} & 0\,0\,3\,0 & 0\,0\,\bar{1}\,0 & 0\,\bar{3}\,0\,0 & \bar{3}\,0\,0\,0 \\
NAF_4(k) = & 1\,0 & 0\,0\,\bar{1}\,0 & 0\,0\,0\,0 & 0\,0\,\bar{5}\,0 & 0\,0\,0\,\bar{3} & 0\,0\,0\,0 & 5\,0\,0\,0 \\
NAF_5(k) = & & 15\,0 & 0\,0\,0\,0 & 0\,0\,\bar{5}\,0 & 0\,0\,0\,\bar{3} & 0\,0\,0\,0 & 5\,0\,0\,0 \\
NAF_6(k) = & & 15\,0 & 0\,0\,0\,0 & \bar{1}\,0\,0\,0 & 0\,0\,\overline{17}\,0 & 0\,0\,0\,0 & \overline{27}\,0\,0\,0
\end{array}
$$

Other Approaches

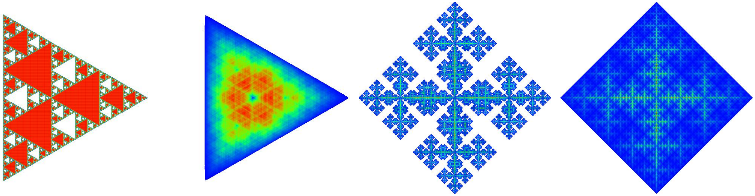
- Double bases systems: $X = \sum_{j=0}^{n_j} \sum_{i=0}^{n_i} x_{i,j} 2^i 3^j$, which give sparse representations.
- Euclidean addition chain systems, inspired of Fibonacci representation: k an integer, we define:
  - $F_1 = 1, F_2 = 2, F_n = F_{n-2} + F_{n-1}$
  - $k = \sum_{i=1}^{n} k_i F_i$ with $k_i = 0, 1$
  - and if $k_i = 1$ then $k_{i\pm 1} = 0$

Conclusions

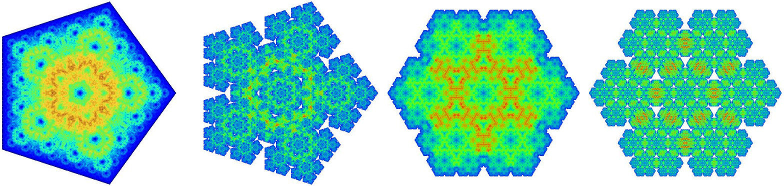
Conclusions

- Now the challenge is to protect against attacks.
  - Redundant Systems different representations for the same value.
  - Leak Resistant Arithmetic in RNS.[13]
  - Fault tolerant arithmetics.
- Lattices and modular arithmetic needs to be more explored.
- A FFT for RNS ?

[13] B. - Imbert - Liardet - Teglia 2004



Thank you!



Annexes

Barret Reduction (1986)

Barrett(A P)

Input $\beta^{n-1} \leq P < \beta^n$ et $A < P^2 < \beta^{2n}$

Output , $R = A \pmod{P}$ et $Q = \lfloor \frac{A}{P} \rfloor$

Core $Q \leftarrow \left\lfloor \frac{\lfloor \frac{\beta^{2n}}{P} \rfloor \times \lfloor \frac{A}{\beta^{n-1}} \rfloor}{\beta^{n+1}} \right\rfloor$

$R \leftarrow A - Q \times P$, $(R < 3P)$

While $R \geq P$ do $R \leftarrow R - P$ and $Q \leftarrow Q + 1$

Complexity : 2 products of $n + 1$ digits

Retour 9

Extension for Q

By the CRT

$$\widehat{Q} = \sum_{i=1}^{n} \left| q_i \left| M_i \right|_{m_i}^{-1} \right|_{m_i} M_i = Q + \alpha M$$

where $0 \leq \alpha < n$.

When $\widehat{Q}$ has been computed it is possible to compute $\widehat{R}$ as

$$\begin{aligned} \widehat{R} = (AB + \widehat{Q}p)M^{-1} &= (AB + Qp + \alpha Mp)M^{-1} \\ &= (AB + Qp)M^{-1} + \alpha p \end{aligned}$$

so that $\widehat{R} \equiv R \equiv ABM^{-1} \pmod{p}$, which is sufficient for our purpose. Also, assuming that $AB < pM$ we find that $\widehat{R} < (n+2)p$ since $\alpha < n$.

(Back 16)

Extension R

Shenoy et Kumaresan (1989):

we have $\left(\sum_{i=1}^{n} M_i \left| |M_i|_{m_i}^{-1} r_i \right|_{m_i}\right) = R + \alpha \times M$

$$\alpha = \left| |M|_{m_{n+1}}^{-1} \left(\sum_{i=1}^{n} \left| M_i \left| |M_i|_{m_i}^{-1} r_i \right|_{m_i} \right|_{m_{n+1}} - |R|_{m_{n+1}} \right) \right|_{m_{n+1}}$$

$$\tilde{r}_j = \left| \sum_{i=1}^{n} \left| M_i \left| |M_i|_{m_i}^{-1} r_i \right|_{m_i} \right|_{\widetilde{m_j}} - |\alpha M|_{\widetilde{m_j}} \right|_{\widetilde{m_j}}$$

(Back 16)

Annexe: Avizienis Algorithm $\boxed{30}$

- We note $S = X + Y$ with
  $X = x_{n-1} \ldots x_0$
  $Y = y_{n-1} \ldots y_0$
  $S = s_n \ldots s_0$
- **Step 1:** For $i = 1$ to n in parallel,

$$
\begin{array}{rcll}
t_{i+1} = & \bar{1} & \text{if, } x_i + y_i < -a + 1 \\
 & 1 & \text{if, } x_i + y_i > a - 1 \\
 & 0 & \text{if, } -a + 1 \leq x_i + y_i \leq a - 1
\end{array}
$$

$$
\begin{array}{rrcl}
\text{and} & w_i & = & x_i + y_i - \beta * t_{i+1} \\
\text{with} & w_n & = & t_0 = 0
\end{array}
$$

- **Step 2:** for $i = 0$ to n in parallel,

$$s_i = w_i + t_i$$

Annexe: NAF_w Computing $\boxed{33}$

Data: Two integers $k \geq 0$ and $w \geq 2$.
Result: $NAF_w(k) = (k_{l-1}k_{l-2}\dots k_1k_0)$.

$l \leftarrow 0$;
while $k \geq 1$ **do**
  if k *is odd* **then**
    $k_l \leftarrow k \mod 2^w$;
    if $k_l > 2^{w-1}$ **then**
      $k_l \leftarrow k_l - 2^w$;
    end
    $k \leftarrow k - k_l$;
  else
    $k_l \leftarrow 0$;
  end
  $k \leftarrow k/2, l \leftarrow l + 1$;
end

Annexe: Double and Add with NAF_w $\boxed{33}$

Data: $P \in E$, $k = \in \mathbb{N}$ et $w \geq 2$, $NAF_w(k) = (k_{l-1}k_{l-2}\dots k_1k_0)$
$P_i = [i]P$ pour $i \in \{1, 3, 5, \dots, 2^{w-1} - 1\}$
Result: $Q = [k]P \in E$.
begin
  $Q \leftarrow P_{k_{l-1}}$;
  pour $i = l - 2 \dots 0$ **faire**
    $Q \leftarrow [2]Q$;
    si $k_i \neq 0$ **alors**
      si $k_i > 0$ **alors**
        $Q \leftarrow Q + P_{k_i}$;
      sinon
        $Q \leftarrow Q - P_{-k_i}$
      fin
    fin
  fin
end

Lattices and Modular Systems

Annexe: Examples of Plantard System $\boxed{28}$

Example1: $P = 53$, $n = 7$, $\beta = 14$, $\rho = 2$.
We have $\beta^7 \equiv 2 \pmod{P}$. In this number system, integers have at least two representations, the total number of representations is 128.
The lattice could be defined by (vectors in row):

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

Lattices and Modular Systems

Annexe: Examples of Plantard System $\boxed{28}$

We can remark that there is a short vector : $(1,1,0,0,0,0,1) = V_6 + 14 * V_5 + 14^2 * V_4 + 14^3 * V_3 + 14^4 * V_2 + (14^5+1) * V_1 + V_7$.
From this vector we can construct a reduced basis of a sublattice, using that: $\beta^7 \equiv 2 \pmod{P}$

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 \\ 2 & 0 & 0 & 0 & 0 & 2 & 1 \end{pmatrix}$$

Lattices and Modular Systems

Annexe: Examples of Plantard System $\boxed{28}$

Example #2: (PhD of Thomas Plantard 2005) The number system must verify: $n = 8,\ \ \beta^8 \equiv 2 \pmod{P}$ and $\rho \sim 2^{32}$.
We search a representation of 2^{32} very sparse giving a large P with $2^{32} \equiv \beta^5 + 1 \pmod{P}$.

We obtain the matrix $M = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{pmatrix}$

We have the lattice $2^{32} Id - M = 0 \bmod P$ thus, P divides $\det\left(2^{32} Id - M\right)$

$P = 115792089021636622262124715160334756877804245386980633020041035952359812890593$

Lattices and Modular Systems

Annexe: Examples of Plantard System $\boxed{28}$

Then β is deduced as a solution of the $\gcd(X^8 - 2, 2^{32} - X^5 - 1)$ modulo P.

$\beta = 144740111277045777827655893952245323141792170589$
$2148839504982773375959039999 6$

The matrix M is useful for the reduction of the coefficients:

$V = 2^{32}V_1 + V_0 = 2^{32}Id.V_1 + V_0 = M.V_1 + V_0$

Here, the reduction if very efficient, two passes could be sufficient.
More generally, M is find with coefficients lower than $2^{k/2}$, which means that three steps are sufficient.

Conversion via CRT $\boxed{16}$

- RNS representation $X = (x_1, x_2, x_3, \cdots, x_n)$
- Shenoy et Kumaresan:

$$\alpha = \left| (M)^{-1}_{m_{n+1}} \sum_{i=1}^{n} \left| M_i \left| \frac{x_i}{M_i} \right|_{m_i} \right|_{m_{n+1}} - |X|_{m_{n+1}} \right|_{m_{n+1}} \quad (1)$$

- Then,

$$X = \sum_{i=1}^{n} M_i \left| \frac{x_i}{M_i} \right|_{m_i} - \alpha M \quad (2)$$

Conversion via Mixed Radix System ??

- RNS representation $X = (x_1, x_2, x_3, \cdots, x_n)$
-
$$\left| \begin{array}{l} a_1 = x_1 \bmod m_1 \\ a_2 = (x_2 - a_1)m_{1,2}^{-1} \bmod m_2 \\ a_3 = ((x_3 - a_1)m_{1,3}^{-1} - a_2)m_{2,3}^{-1} \bmod m_3 \\ a_4 = (((x_4 - a_1)m_{1,4}^{-1} - a_2)m_{2,4}^{-1}) - a_3)m_{3,4}^{-1} \bmod m_4 \\ \vdots \\ a_n = (\cdots(x_n - a_1)m_{1,n}^{-1} - a_2)m_{2,n}^{-1}) - \cdots - a_{n-1})m_{n-1,n}^{-1} \bmod m_n \end{array} \right.$$
with $m_{i,j}^{-1}$ inverse of m_i modulo m_j
- Mixed Radix representation $X = (a_1, a_2, a_3, \cdots, a_n)$
- $X = a_1 + a_2 m_1 + a_3 m_1 m_2 + \cdots + a_n m_1 \cdots m_{n-1}$