A Residue Approach to the Finite Field Arithmetics

JC Bajard

LIRMM, CNRS UM2
161 rue Ada, 34392 Montpellier cedex 5, France

CIRM 2009

Contents

Introduction to Residue Systems

Introduction to Residue Systems

- In some applications, like cryptography, we use finite field arithmetics on huge numbers or large polynomials.
- Residue systems are a way to distribute the calculus on small arithmetic units.
- Are these systems suitable for finite field arithmetics?

Residue Number Systems in $\mathbb{F}_p$, p prime

- Modular arithmetic mod p, elements are considered as integers.
- 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 product, but an extension of the base could be necessary and a reduction modulo p is needed.

Residue Number Systems in $\mathbb{F}_p$, p prime

- $\Phi(m) = \sum_{\substack{p \leq m \\ p \text{ prime}}} \log p = \log \prod_{\substack{p \leq m \\ p \text{ prime}}} p \sim m$
- 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 $\Theta(f(\log m))$.

Lagrange representations in $\mathbb{F}_{p^k}$ with $p > 2k$

- Arithmetic modulo $I(X)$, an irreducible $\mathbb{F}_p$ polynomial of degree k. Elements of $\mathbb{F}_{p^k}$ are considered as $\mathbb{F}_p$ polynomials of degree lower than k.
- Lagrange representation
  - is defined by k different points $e_1, \ldots e_k$ in $\mathbb{F}_p$. $(k \leq p.)$
  - A polynomial $A(X) = \alpha_0 + \alpha_1 X + \ldots + \alpha_{k-1} X^{k-1}$ over $\mathbb{F}_p$ is given in Lagrange representation by:

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

  - Remark: $a_i = A(e_i) = A(X) \bmod (X - e_i)$. If we note $m_i(X) = (X - e_i)$, we obtain a similar representation as RNS.
- Operations are made independently on each $A(e_i)$ (like in FFT or Tom-Cook approaches). We need to extend to $2k$ points for the product.

Trinomial residue in $\mathbb{F}_{2^n}$

- Arithmetic modulo $I(X)$, an irreducible $\mathbb{F}_2$ polynomial of degree n. Elements of $\mathbb{F}_{2^n}$ are considered as $\mathbb{F}_2$ 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 for each $m_i(X)$

Residue Systems

- Residue systems could be an issue for computing efficiently the product.
- The main operation is now the modular reduction for constructing the finite field elements.
- The choice of the residue system base is important, it gives the complexity of the basic operations.

Modular reduction in Residue Systems

Reduction of Montgomery on $\mathbb{F}_p$

- The most used reduction algorithm is due to Montgomery (1985)[9]
- For reducing A modulo p,
  one evaluates $q = -(Ap^{-1}) \bmod 2^s$,
  then one constructs $R = (A + qp)/2^s$.
  The obtained value satisfies: $R \equiv A \times 2^{-s} \pmod{p}$ and $R < 2p$ if $A < p2^s$.
  We note $\mathrm{Montg}(A, 2^s, p) = R$.
- Montgomery notation: $A' = A \times 2^s \bmod p$
  $\mathrm{Montg}(A' \times B', 2^s, p) \equiv (A \times B) \times 2^s \pmod{p}$

Residue version of Montgomery Reduction

- The residue base is such that $p < M$ (or $\deg M(X) \geq \deg I(X)$)
- We use an auxiliary base such that $p < M'$ (or $\deg M'(X) \geq \deg I(X)$), M' and M coprime. (Exact product, and existence of M^{-1})
- 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.

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

The extension comes from the Lagrange interpolation.
If $(a_1, ..., a_k)$ is the residue representation in the base M, then

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

The factor α can be, in certain cases, neglected or computed.[1]
Another approach consists in the Newton interpolation where A is correctly reconstructed. [4]
In the polynomial case, the term $-\alpha M$ vanishes.

Extension for Q

By the CRT

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

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 $\alpha < n$.

Extension R

Shenoy et Kumaresan (1989):

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

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

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

Extension of Residue System Bases

We first translate in an intermediate representation (MRS):

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

We evaluate A, with Horner's rule, as

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

Features of the residue system

- Efficient multiplication, the cost being the cost of one multiplication on one residue.
- Costly reduction: $O(k^{1.6})$ for trinomials [4], $2k^2 + 3k$ for RNS [1], $O(k^2)$ for Lagrange representation [5].
- If we take into account that most of the operations are multiplications by a constant, the cost can be considerably smaller.

Applications to Cryptography

Elliptic curve cryptography

- The main idea comes from the efficiency of the product and the cost of the reduction in Residue Systems.
- We try to minimize the number of reductions. A reduction is not necessary after each operation. Clearly, for a formula like $A \times B + C \times D$, only one reduction is needed.
- Elliptic Curve Cryptography is based on points addition. We use appropriate forms (Hessian, Jacobi, Montgomery...) and coordinates: projective, Jacobian or Chudnowski...
- For 512 bits values Residues Systems for curves defined over a prime field, are more efficient than classical representations.[2]

Pairings

- To summarize we define a pairing as follows: let G_1 and G_2 be two additive abelian groups of cardinal n and G_3 a multiplicative group of cardinal n.
- A pairing is a function $e : G_1 \times G_2 \longrightarrow G_3$ which verifies the following properties: Bilinearity, Non-degeneracy.
- For pairings defined on an elliptic curve E over a finite field $\mathbb{F}_p$, we have $G_1 \subset E(\mathbb{F}_p)$, $G_2 \subset E(\mathbb{F}_{p^k})$ and $G_3 \subset \mathbb{F}p^k$, where k is the smallest integer such that n divides $p^k - 1$, k is called the embedded degree of the curve.

Pairings

- The construction of the pairing involves values over $\mathbb{F}_p$ and $\mathbb{F}_{p^k}$ into the formulas. An approach with Residue Systems, similar to the one made on ECC could be interesting.[3]
- k is most of the time chosen as a small power of 2 and 3 for algorithmic reasons. Residue arithmetics allow to pass over this restriction.
- With pairings, we can also imagine two levels of Residue Systems: one over $\mathbb{F}_p$ and one over $\mathbb{F}p^k$.

Conclusion on Residue Systems

Conclusion

- If your number system is not efficient, then it remains to try the residues.

- Bajard, J.C., Didier, L.S., Kornerup, P.: Modular multiplication and base extension in residue number systems. 15th IEEE Symposium on Computer Arithmetic, 2001 Vail Colorado USA pp. 59–65
- Bajard, J.C., Duquesne, S., Ercegovac M. and Meloni N.: Residue systems efficiency for modular products summation: Application to Elliptic Curves Cryptography, in Advanced Signal Processing Algorithms, Architectures, and Implementations XVI, SPIE 2006, San Diego, USA.
- Bajard, J.C. and ElMrabet N.: Pairing in cryptography: an arithmetic point of view, Advanced Signal Processing Algorithms, Architectures, and Implementations XVII, part of the SPIE Optics & Photonics 2007 Symposium. August 2007 San Diego, USA.

- J.C. Bajard, L. Imbert, and G. A. Jullien: Parallel Montgomery Multiplication in $GF(2^k)$ using Trinomial Residue Arithmetic, 17th IEEE symposium on Computer Arithmetic, 2005, Cape Cod, MA, USA.pp. 164-171
- J.C. Bajard, L. Imbert et Ch. Negre, Arithmetic Operations in Finite Fields of Medium Prime Characteristic Using the Lagrange Representation, journal IEEE Transactions on Computers, September 2006 (Vol. 55, No. 9) p p. 1167-1177
- Bajard, J.C., Meloni, N., Plantard, T.: Efficient RNS bases for Cryptography IMACS'05, Applied Mathematics and Simulation, (2005)
- Garner, H.L.: The residue number system. IRE Transactions on Electronic Computers, EL **8:6** (1959) 140–147
- Knuth, D.: Seminumerical Algorithms. The Art of Computer Programming, vol. 2. Addison-Wesley (1981)

- Montgomery, P.L.: Modular multiplication without trial division. Math. Comp. **44:170** (1985) 519–521
- Svoboda, A. and Valach, M.: Operational Circuits. Stroje na Zpracovani Informaci, Sbornik **III**, Nakl. CSAV, Prague, 1955, pp.247-295.
- Szabo, N.S., Tanaka, R.I.: Residue Arithmetic and its Applications to Computer Technology. McGraw-Hill (1967)