New Residue Arithmetic Based Barrett Algorithms: Modular Polynomial Computations

Hari Krishna Garg
Electrical & Computer Engineering Department
National University of Singapore
elegkh@nus.edu.sg

Hanshen Xiao
Department of Mathematics
Tsinghua University
hxsiao@mit.edu

Abstract—We derive a new computational algorithm for Barrett technique for modular polynomial multiplication, termed BA-P. Residue arithmetic is applied to BA-P to obtain a new Barrett algorithm for modular polynomial multiplication (BA-MPM). The work is focused on an algorithm that carries out computation using modular arithmetic without conversion to large degree polynomials. There are several parts to this work. First, we set up a new BA-P using polynomials other than \( u^p \). Second, residue arithmetic based BA-MPM is described. A complete mathematical framework is described including proofs for the results. Third, we present a computational procedure for BA-MPM. Fourth, the BA-MPM is used as a basis for algorithms for modular polynomial exponentiation (MPE). Applications are in areas of signal security and cryptography.

Keywords—Barrett Algorithm (BA), BA for Polynomials (BA-P), Modular Polynomial Multiplication (MPM), Montgomery Multiplication (MM), Residue Polynomial Systems (RPS), Chinese Remainder Theorem for Polynomials (CRT-P), BA-P based on MPM (BA-MPM), Modular Polynomial Exponentiation (MPE), Base Extension for Polynomials (BEX-P).

I. INTRODUCTION

Cryptography techniques play an important role in the security of electronic systems. Instances of such cryptography techniques include RSA (Rivest-Shamir-Adleman), Rabin, Diffie Hellman and El Gamal. These techniques deal with arithmetic defined in a large size finite fields GF(\( p \)) and/or GF(\( p^n \)), where \( p \) is a prime integer and \( N \) is an integer. A large size field may be realized by setting \( p = 2 \) (binary arithmetic) and \( N \) to be a large value. Here, we also deal with finite fields with large values of \( N \), say 500 to 5,000. The elements in GF(2\( ^N \)) are expressed as polynomials over GF(2) of degree up to \( N \) – 1. A challenge is to perform the following computations efficiently:

1. Multiplication of two elements in GF(2\( ^N \)):
   \[ C(u) = A(u) \cdot B(u) \pmod{P(u)}; \deg(P(u)) = N; \]
2. Modular exponentiation in GF(2\( ^N \)):
   \[ C(u) = A(u)^k \pmod{P(u)}; \deg(P(u)) = N. \]

Here, \( P(u) \) is an irreducible (or primitive) polynomial in GF(2). This eliminates use of Chinese remainder theorem for polynomials (CRT-P) to compute \( C(u) \). Computations in 1 and 2 without mod \( P(u) \) are simpler while mod \( P(u) \) computation is challenging. Also modular polynomial exponentiation (MPE) is computed via repeated use of modular polynomial multiplication (MPM). Hence, an efficient algorithm must be used for MPM. In many situations, \( N \) is large.

Residue arithmetic is used to express a large size ring as a direct product of a number of smaller size rings. Residue number systems (RNS) have been applied in Barrett algorithm (BA) and Montgomery multiplication (MM), to compute modular operations in large size integer rings. However, there is a distinct gap when it comes to residue Polynomial Systems (RPS) based BA and MM.

Contributions of the work are as follows. The primary objective is to compute MPM and MPE efficiently for applications in signal security and cryptography. We first describe a new BA for modular polynomial multiplication (BA-P) for computing the quotient \( C(u) \) associated with \( X(u) \) when it is divided by \( P(u) \). It is assumed that \( N = \deg(P(u)) \) is a large integer. Second, a residue arithmetic based BA-P, termed BA-MPM, is described for modular polynomial multiplication. Third, a computationally efficient procedure for the new BA-MPM is described. Fourth, the new BA-MPM is used as a basis for MPE. The results are general and valid for all fields such as GF(\( p^n \)), rational, real, and complex numbers.

There is an abundance of research on MM and BA [1]-[13]. Polynomial versions of MM and BA can be found in [14]-[22]. A digit-serial multiplication in GF(2\( ^p \)) based on Barrett modular reduction is presented in [15]. A version of digit-serial multiplication algorithm is described in [16]. Other aspects are explored in [19]. Further details of BA and MM are available in [23]-[28].

However, there is no paper on using residue arithmetic to compute MPM and MPE via BA. It is this particular aspect that we deal with in this paper. The algorithm described here begins with reformulating BA such that the new BA-P stays within residue arithmetic.

The organization of this paper is as follows. Section II provides mathematical preliminaries on arithmetic, BA-P, RPS, CRT-P, and base extension for polynomials (BEX-P). The new BA-P is described in Section III. The computational steps for RPS based BA-P algorithm are presented in Section IV. Examples are presented to illustrate the algorithm. In Section V, we describe an algorithm for MPE that uses the new BA-MPM. Section VI is on conclusions.

II. MATHEMATICAL PRELIMINARIES

Polynomial Arithmetic. Given \( X(u) \) and \( A(u) \) with coefficients in a field \( F \), consider dividing \( X(u) \) by \( A(u) \) to write
\[
X(u) = Q(u) \cdot A(u) + R(u).
\]

Here, \( Q(u) \) is quotient and \( R(u) \) is remainder. Also, \( \deg(R(u)) < \deg(A(u)) \) with \( \deg(Q(u)) = \deg(A(u)) - \deg(A(u)) \). We write (1) as
\[
X(u) = R(u) \pmod{A(u)}.
\]

Dividing both sides of (1) by \( A(u) \), we get,
\[
X(u) / A(u) = Q(u) + R(u) / A(u).
\]

The last term on the right when expressed as a sum of powers of \( u \) will only contain negative powers. We also write
\[
Q(u) = \langle u \rangle \cdot X(u) / A(u) \pmod{A(u)} = \langle u \rangle \cdot Y(u),
\]

where \( \langle u \rangle \) is the floor function of \( Y(u) \). \( Q(u) \) and \( R(u) \) are unique. This process can be repeated between \( Q(u) \) and \( A(u) \). Thus
\[
Q(u) = Q(u) \cdot A(u) + R(u),
\]

\[
0 \leq \deg(R(u)) < \deg(A(u)).
\]

A generalization of (5) leads to
\[
X(u) = Q(u) \cdot (A(u) - A(u)) + R(u) \cdot (A(u) - A(u)) + \ldots + R(u) \cdot A(u) + A(u).
\]
This expression is useful when involving RPS and BEX-P.

**Barrett Algorithm for Polynomials (BA-P).** Given $A(u), B(u),$ and $P(u)$ in $\mathbb{GF}(p)$, $\deg(A(u)), \deg(B(u)) < \deg(P(u))$, NMPM computes:

$$C(u) = A(u) \cdot B(u) \mod P(u).$$

Let $X(u) = A(u) \cdot B(u)$. BA-P computes quotient $Q(u)$ such that $X(u) = Q(u) \cdot P(u) + C(u)$. Then $C(u)$ is computed as $C(u) = X(u) - Q(u) \cdot P(u)$. Given $X(u)$ and $P(u)$, BA-P expresses $Q(u)$ as

$$Q(u) = \lfloor X(u) / P(u) \rfloor.$$

In the current versions of BA-P [14]-[22], (8) is computed as $Q(u) = \lfloor X(u) / u^{a} \cdot \Omega(u) / u^{b} \rfloor \equiv \lfloor X(u) / u^{a} \rfloor \cdot \lfloor \Omega(u) / u^{b} \rfloor \mod 1$.

Where $\lfloor \Omega(u) \rfloor$ is pre-computed as $\lfloor \Omega(u) \rfloor = \lfloor u^{b} \cdot P(u) \rfloor$. Scalars $a$ and $b$ are chosen such that $Q(u)$ in (9) is same as $Q(u)$ in (8) [15, 16]. BA-P consists of steps:

**Residue Polynomial System (RPS) [23, 29].** A RPS defined mod $M(u)$ is a ring defined by $n$ co-prime polynomials $M_{1}(u), M_{2}(u), \ldots, M_{n}(u)$ with elements in field $F$. The elements in RPS are polynomials of degree up to $L - 1, L = \deg(M(u))$, where

$$M(u) = \sum M_{i}(u).$$

A polynomial $X(u)$ in the RPS is represented as $n$ residues,

$$X(u) \leftrightarrow X(u) \leftrightarrow [X(u) \mod M_{1}(u) \ldots X(u) \mod M_{n}(u)],$$

where $X(u) = X(u)$ (mod $M(u)$), $i = 1, 2, \ldots, n$.

**Chinese remainder theorem for polynomials (CRT-P) [2, 3, 23, 29].** Given $X(u), X(u), \deg(X(u)) < L$, is computed via CRT-P, as

$$X(u) \equiv \sum_{i=1}^{n} T_{i}(u) \cdot X_{i}(u) \mod M_{i}(u).$$

Polynomials $T_{i}(u), \deg(T_{i}(u)) < \deg(M(u))$, are computed a-priori via

$$T_{i}(u) = \begin{cases} \begin{array}{c} M_{i}(u) \\mod M_{i}(u), \\ i = 1, 2, \ldots, n. \end{array} \end{cases}$$

CRT-P computation of $X(u)$ involves large degree polynomials.

**Base Extension for Polynomials (BEX-P).** Consider $X(u)$, residues of $X(u)$ in (11). BEX-P consists in computing $t$ additional residues of $X(u), X(u) = (X(u) \mod M_{j}(u)), j = n + 1, \ldots, n + t$, in a RPS defined mod $M(u), M_{j}(u) = \prod_{j=1}^{n} M_{j}(u)$, where gcd($M(u), M_{j}(u)$) = 1. BEX-P is intense computationally. Using (12), we compute it as [22]:

$$X_{j}(u) = \sum T_{i}(u) \cdot X_{i}(u) \mod M_{j}(u) \cdot \frac{M(u)}{M_{j}(u)},$$

$$j = n + 1, \ldots, n + t.$$

III. A NEW BARRETT ALGORITHM FOR POLYNOMIALS

A RPS based Montgomery multiplication algorithm has been described in [22]. However, there is no such algorithm for the BA-P. We cite [15, 16, 19] and the references therein. They have used modulo polynomials of the type $u^{a}$. Clearly, this doesn’t lend itself to RPS. Here, we first revisit the computation of $Q(u)$ in (8). We now introduce two polynomials $G(u)$ and $H(u)$, not necessarily of the form $u^{a}$, and approximate $Q(u)$ in (8) as

$$Q(u) = \lfloor X(u) / G(u) \cdot \Omega(u) / H(u) \rfloor \equiv \lfloor X(u) / G(u) \rfloor \cdot \lfloor \Omega(u) / H(u) \rfloor \mod 1.$$ (14)

$\mu(u)$ is pre-computed as $\mu(u) = \lfloor \Omega(u) \rfloor \cdot \lfloor G(u) \cdot H(u) / P(u) \rfloor$.

Since $X(u) = A(u) \cdot B(u), \deg(A(u)), \deg(B(u)) < N, \deg(X(u)) \leq 2 \cdot N - 2$.

Now we derive conditions on $G(u)$ and $H(u)$ for approximation of $Q(u)$ in (14) to be equal to $Q(u)$ in (8). Let $V(u)$ is polynomial part of $F(u)$ consisting of terms with positive powers of $u$. Consider dividing $V(u)$ by $S(u)$ to write $V(u) = Q(u) \cdot S(u) + R(u)$. Then we have

$$V(u) / S(u) = Q(u) + R(u) / S(u) = Q(u) / S(u) + \delta(u), \deg(\delta(u)) \leq -1.$$ (15)

Applying (15) to (14), we get

$$Q(u) = \frac{X(u)}{G(u)} + \frac{\delta(u)}{H(u)} + \frac{G(u) \cdot H(u)}{P(u)} \cdot \mu(u).$$ (16)

Here, $\deg(\delta(u))$, $\deg(\delta(u)) \leq -1$. We wish the second term in the above summation to have degree less than 0. To achieve that,

(A) $\deg(\frac{X(u)}{G(u)}) \leq \deg(H(u))$;

(B) $\deg\left(\frac{G(u) \cdot H(u)}{P(u)}\right) \leq \deg(H(u))$.

We assume these conditions to be satisfied. Thus (16) becomes:

$$Q(u) = \left[\frac{X(u)}{G(u)} + \frac{\delta(u)}{H(u)} + \frac{G(u) \cdot H(u)}{P(u)} \cdot \mu(u)}{H(u)}\right].$$ (17)

as $\deg(\delta(u)) \leq -1$. We note that $\deg(A(u)) = \deg(G(u)) + \deg(H(u)) - \deg(P(u))$. This analysis leads to the following theorem:

**Theorem 1.** Let $A(u), B(u)$ and $P(u)$ be given such that $0 \leq \deg(A(u)), \deg(B(u)) < \deg(P(u))$. For the computation $X(u)$ mod $P(u), X(u) = A(u) \cdot B(u)$, if $G(u)$ and $H(u)$, $\alpha = \deg(G(u))$ and $\beta = \deg(H(u))$, satisfy the conditions

$$\deg(X(u)) \leq 2 \cdot N - 2 \leq \alpha + \beta; \alpha \leq \deg(P(u)) = N.$$ (18)

then $Q(u)$ in (17) is same as the quotient $\lfloor X(u) / P(u) \rfloor$.

A generalization of $G(u)$ and $H(u)$ from polynomials of the type $u^{a}$ is crucial. A choice of degrees that satisfy (18) is $\alpha = \beta = N$. $G(u)$ and $H(u)$ can be identical. This analysis leads to the following new BA-P:

**A New Barrett Algorithm for $A(u) \cdot B(u) \mod P(u)$ (BA-P)**

Input: $A(u), B(u), P(u), G(u), H(u); 0 \leq \deg(A(u)), \deg(B(u)) < N, N = \deg(P(u))$, $\alpha = \deg(G(u)), \beta = \deg(H(u))$.

Output: $Q(u)$ (quotient when $A(u) \cdot B(u)$ is divided by $P(u)$).

Step 0. Pre-compute $\mu(u) = \lfloor \Omega(u) \cdot H(u) / P(u) \rfloor, \deg(\mu(u)) = \alpha + \beta - N$ (one-time)

Compute
Step 1. \( X(u) = A(u) \cdot B(u), \deg(X(u)) \leq 2 \cdot N - 2 \) (ordinary mult)

Step 2. \( D(u) = \left\lfloor \frac{X(u)}{G(u)} \right\rfloor \), \( \deg(D(u)) \leq 2 \cdot N - 2 \) (ordinary mult)

Step 4. \( Q(u) = \left\lfloor \frac{E(u)}{H(u)} \right\rfloor \), \( \deg(Q(u)) \leq N - 2 \) (quotient)

Once \( Q(u) \) is computed, remainder \( X(u) \mod P(u) \) is computed as

Step 5. \( C(u) = X(u) - Q(u) \cdot P(u), \deg(C(u)) \leq N - 1 \) (ordinary mult)

The conditions in (18) required for \( G(u) \) and \( H(u) \) are general and open door to a range of possibilities for different computational steps.

Example 1. Assume that the computation is defined in GF(2). Let \( N = 6 \), \( P(u) = u^6 + u + 1 \). We can choose \( G(u) = u^6 + 1, H(u) = u^6 + 1 \). Then \( \mu(u) = u^6 + u + 1 \) and \( \lambda(u) = u^6 + u^2 + 1 \). Let \( X(u) = u^6 + u^3 + u^2 + u + 1 \). We have \( D(u) = u^6 + u^3 + u^2 + u^2 + u^2 + u^2 \). \( Q(u) = u^6 + u^3 + u^2 + u^2 + u^2 \). We use \( C(u) = P(u) - Q(u) \cdot D(u) \). A trivial solution to \( \left\lfloor \frac{X(u)}{H(u)} \right\rfloor = 2 \) is available in residues for \( M(u) \) for \( G(u) \). Hence, we need BEX-P to expand quotient residues back to mod \( M(u) \). Similarly, we need BEX-P to expand quotient residues of \( \left\lfloor \frac{E(u)}{H(u)} \right\rfloor \) computed mod \( M(u) \) to mod \( M(u) \). Such a BEX-P algorithm is described in Section II.

A RPS based new Barrett Algorithm for Polynomials (BA-MPM)

Given: \( M(u), G(u), H(u) \). Let \( M(u) \) have \( n \) factors.

In step 2a, first \( a \) factors of \( M(u) \) give \( G(u) \); and in step 4a, first \( b \) factors of \( M(u) \) give \( H(u) \). There is no loss in generality.

Input: Residues of \( A(u) \) and \( B(u) \). \((A(u), B(u)) = (A(u), B(u)) \mod M(u), i = 1, ..., n \).

Pre-computational Step:

Step 0. Compute \( \mu(u), i = 1, ..., n, \mu(u) = \left\lfloor \frac{G(u) \cdot H(u)}{P(u)} \right\rfloor \).

Computational Steps:

Step 1. Modulo mult. \( X(u) = A(u) \cdot B(u), i = 1, ..., n \).

Step 2a. Quotient. Quotient residues \( D(u), i = a + 1, ..., n \), from residues \( X(u), i = 1, ..., n \), and moduli \( G(u), i = 1, ..., a \).

Step 2b. BEX-P. Use BEX-P on \( D(u) \), \( i = a + 1, ..., n \), to get a residues \( D(u), i = 1, ..., a \).

Step 3. Modulo mult. \( E(u) = D(u) \cdot \mu(u), i = 1, ..., n \).

Step 4a. Quotient. Quotient residues \( Q(u), i = b + 1, ..., n \), from residues \( E(u), i = 1, ..., n \), and moduli \( H(u), i = 1, ..., b \).

Step 4b. BEX-P. Use BEX-P on \( Q(u) \), \( i = b + 1, ..., n \), to get b residues \( Q(u), i = 1, ..., b \).

Step 5. Remainder. \( C(u) = X(u) - Q(u) \cdot P(u), i = 1, ..., n \).

Provided that \( L - B > N - 1 \) (a rather trivial condition at this stage), steps 4b and 5 may also be swapped. In that case, we have:

Step 5. Remainder. \( C(u) = X(u) - Q(u) \cdot P(u), i = b + 1, ..., n \).

Step 6. BEX-P. Use BEX-P on residues \( C(u), i = b + 1, ..., n \), to get b residues \( C(u), i = 1, ..., b \).

Example 3. Let the computation be defined in GF(2). Let \( N = 2^n - 1 \), and \( \alpha = \beta = N \). In this case, \( \deg(M(u)) > 2 \cdot N - 2 \). We choose \( M(u) = (u^6 - 1), \left\lfloor \frac{u^{n+2} - 1}{u - 1} \right\rfloor \) with \( \deg(u) = u^6 - 1 \). Also, \( \gcd(a^n - 1, a - 1) = a^{\gcd(a - 1)} - 1 \). In our case, \( N \) and \( N + 2 \) are two consecutive odd integers, hence \( \gcd(N, N + 2) = 1 \). Thus, \( u^{n+2} - 1 \) and \( \left\lfloor \frac{u^{n+2} - 1}{u - 1} \right\rfloor \) are co-prime.
of degree 10 or less. Similarly, $u^{1025} - 1$ has factors of degree 20 or less. Let $M(u) = (u^{1023} - 1) \cdot (u^{1025} - 1)$, we assume that $\deg(A(u)) = \deg(B(u)) = N = 1022$. If $A(u)$ and $B(u)$ have degree less than 1022, then we pad them with 0’s and treat them as polynomials of degree 1022. Thus, $A(u) \cdot B(u)$ mod $P(u)$, $N = 1023$, is computed using arithmetic where half the module have degree up to 10 and half have degree up to 20.

**Example 5.** Consider $A(u) \cdot B(u)$ mod $P(u)$ in the field of real or complex numbers. We let $G(u) = H(u) = u^N - 1$ and $M(u) = u^N - 1 = (u^N - 1) \cdot (u^N + 1)$. Let $\omega$ be the 2 complex roots of unity. In this case, computations use DFT and IDFT via FFT. For instance, $X(u)$ in step 1 can be computed using a size 2 DFT. In step 2a, let $R(u)$ denote the remainder $X(u) \mod G(u)$. Then $R(u)$ is computed as a size N IDFT of the even DFT coefficients of $X(u)$. We write $D(u) = [X(u) - R(u)] / (u^N - 1)$.

Substituting odd powers of $u$, we get $D(u) = [X(u) - R(u)] \mod G(u)$. The computation of $R(u)$ in step 2a is first taking size $N$ IDFT of $D(u)$, $k = 0, \ldots, N - 1$, obtaining the sequence $D_{k} \mod \omega$, $i = 0, \ldots, N - 1$. The computation of $D(u)$ in step 2a is performed in parallel.

**APPENDIX: Computing Quotient Residues in RPS**

**Problem.** Given residues $X(u)$ of $X(u)$, $X(u) = X(u) \mod M(u)$, $i = 1, \ldots, n$, $M(u) = M(u) \cdot M(u)$, $M(u) = \prod_{i=1}^{n} M_{i}(u)$, $M(u) = \prod_{i=1}^{n} M_{i}(u)$, $\gcd(M(u), M(u)) = 1$, compute residues of quotient $Q(u)$ when $X(u)$ is divided by $M(u)$, $\deg(Q(u)) < \deg(M(u))$.

We revisit polynomial arithmetic described in Section II and use it to get an algorithm for computing residues of $Q(u)$. Consider (1) when $X(u), A(u)$ and $R(u)$ are known. We compute $Q(u)$ as $Q(u) = (\chi(u) - R(u)) \cdot A(u)^{-1}$. When $Q(u), A(u)$ and $R(u)$ are known, we compute $Q(u)$ as $Q(u) = (\chi(u) - R(u)) \cdot A(u)^{-1}$.

This is carried out recursively to finally compute $Q(u)$ as $Q(u) = (\chi(u) - R(u)) \cdot A(u)^{-1}$.

The representation of $X(u) \mod$ (6) is valid. It is reproduced below:

$X(u) = X(u) \cdot [A(u) - A(u)] + R(u) \cdot [A(u) - R(u)]$.

We apply the arithmetic in (1) (A4) to RPS defined mod $M(u)$. Given modular $M(u)$ and residues $X(u)$, we set $A(u) = M(u)$.

Thus, $R(u) = X(u)$. This leads to $Q(u) = (\chi(u) - R(u)) \cdot M(u)^{-1}$.

Since $X(u)$ is expressed in terms of its residues and $M(u)$ exists only mod $M(u)$, $i = 2, \ldots, n$, we compute residues of $Q(u)$ in (A6) by taking mod $M(u)$, $i = 2, \ldots, n$, of both sides. Thus, $Q(u) = (\chi(u) - X(u)) \cdot M(u)^{-1}$ (mod $M(u)$), $i = 2, \ldots, n$.

Again, $\deg(Q(u)) < \deg(M(u))$, $\deg(Q(u)) < \deg(M(u))$.

Thus, $Q(u)$ is uniquely expressed by its residues $Q(u)$, $i = 2, \ldots, n$. After the first iteration in (A1), $R(u) = X(u)$ (mod $M(u)$) = $Q(u)$.

**VI. CONCLUSIONS**

In this work, new Barrett algorithms are described for computing $A(u) \cdot B(u)$ mod $P(u)$ and $A(u)^{2}$ mod $P(u)$, $P(u)$ being an irreducible polynomial of degree $N$. A residue polynomial system based new Barrett algorithm is described that uses only residue arithmetic thus avoiding large degree polynomial multiplication that may be computationally intensive. All the algorithms as described here are a first. The previously known Barrett algorithms use powers of $u$ to scale the various computations.

**ACKNOWLEDGMENT**

The work of H.K. Garg is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the Interactive Digital Media Programme Office at the NUS-ZJU Sensor-Enhanced Social Media (SeSaMe) Centre. The work of Hanshen Xiao is supported by Initiative Scientific Research Pro-gram of Tsinghua University under Grant No. 2015THZQ & 20161080166.
REFERENCES