Barrett Design

Zeyu Xie

April 22, 2021

Contents

1 Introduction

PUBLIC-KEY cryptography (PKC) becomes important based on frequent digital communication systems nowadays. However, it is still a challenge to implement PKC efficiently due to big data.

2 Basic Barrett Design

Some notation is given to make the following discussion easier. We define an N-bit integer X in radix r representation as X = $(X_{n_m-1}...X_0)_r$, where r = 2^m, $n_m = \lceil N/m \rceil$ and an N-bit integer Y in the same way; We refer to n_m as digit count and m as one digit width. Also, a special case is when m = 1, the representation of X = $(X_{n-1}...X_0)_2$ is called bit representation.

2.1 Basic Modualr Multiplication

Here, we define an intermediate variable T_i and an intermediate result Z_i.

Algorithm 1 Basic Modular Multiplication

Input: X,Y ∈ [0,M), $2^N - 1 \leq M < 2^N$, r = 2^m, $n_m = \lceil N/m \rceil$
Output: $Z \in [0, M)$
1: initial $Z_{n_m} = 0$
2: **for** i = n_m - 1 **to** 0 **do**
3: $T_i = Z_{i+1}$r + XY_i
4: $q_i = \lfloor T_i/m \rfloor$
5: $Z_i = T_i$ - q_iM
6: **end**
7: Z = Z_0
8: **return** Z

Proof: Z_i is in the range [0,M)

for simple modular multiplication: Z = XY mod M.

In every intermediate result Z_i:

$Z_i = T_i$ - $\lfloor T_i/M \rfloor M$.

$Z/M = T_i/M$ - $\lfloor T_i/M \rfloor \in [0, 1)$ due to the characteristic of floor function.

Hence, we can conclude that $Z_i \in [0, M)$

2.2 Classic Barrett Modualr Multiplication

Due to the expensive cost of division in hardware implementation, It shows the classic Barrett Modualr Multiplication (BMM) as below, instead of using division operation, we replace to multiplication and shift, where shift operation is simple for hardware implementation. To better explain the algorithm, we define α and β to control error.

Algorithm 2 Classic Barrett Modular Multiplication

Input: X,Y $\in$ [0,M), $2^N - 1 \leq M < 2^N$, r = 2^m, n$_m$ = $\lceil N/m \rceil$, $u = \lceil 2^{N+\alpha}/M \rceil$
Output: $Z \in [0, M)$
1: initial Z$_{n_m}$ = 0
2: **for** i = n_m - 1 **to** 0 **do**
3: T_i = Z_{i+1}r + XY_i
4: $q_i = \lfloor \lfloor \frac{T_i}{2^{N+\beta}} \rfloor \mu / 2^{\alpha-\beta} \rfloor$
5: Z_i = T_i - q_iM
6: **end**
7: Z = Z_0
8: **if** $Z \geq M$ **then**
9: Z = Z - M (correction step)
10: **return** Z

Proof: Z_i is in the range [0,2M)
After replacing the division operation to multiplication and shift shown at step 4, now the quotient q_i is related with three floor function instead of one, hence, it is possible that the intermediate result Z lies in a bigger range.

The error is produced by three floor function, so let us introduce these three differences e_1, e_2 and e_3:

$e_1 = \frac{T_i}{2^{N+\beta}}$ - $\lfloor \frac{T_i}{2^{N+\beta}} \rfloor$

$e_2 = \frac{2^{N+\alpha}}{M}$ - $\lfloor \frac{2^{N+\alpha}}{M} \rfloor$

$e_3 = \frac{\lfloor \frac{T_i}{2^{N+\beta}} \rfloor \mu}{2^{\alpha-\beta}}$ - $\lfloor \frac{\lfloor \frac{T_i}{2^{N+\beta}} \rfloor \mu}{2^{\alpha-\beta}} \rfloor$

Apparently, these three differences e_1,e_2,$e_3 \in$ [0,1) due to the characteristic of floor function.

Then, we can calculate the expression of q_i:

$q_i = \frac{T_i}{M}$ - $\frac{2^{N+\beta}}{M} e_1$ - $\frac{T_i}{2^{N+\alpha}} e_2$ - e_3 + $\frac{e_1 e_2}{2^{\alpha-\beta}}$

To get the final error for intermediate result Z_i, we modify the above equation:

$\frac{T_i}{M}$ - $q_i = e = \frac{2^{N+\beta}}{M} e_1 + \frac{T_i}{2^{N+\alpha}} e_2 + e_3$ - $\frac{e_1 e_2}{2^{\alpha-\beta}}$

For $\alpha \geq m+3$ and $\beta \leq -2$, we assuming $Z_{i+1} \in [0, 2M)$, then following the algorithm and do error analysis:

$\frac{2^{N+\beta}}{M} e_1 \leq \frac{2^{N-2}}{2^{N-1}} e_1 \leq 0.5$

$\frac{T_i}{2^{N+\alpha}} e_2 < \frac{3 \cdot 2^{N+m}}{2^{N+m+3}} e_2 < 0.375$

$e_3 < 1$

Hence, we can conclude that $e \in [0, 2)$ and $Z_i = T_i - q_i M = eM \in [0, 2M)$.

2.3 Proof of whole process

1: floor function: $q = \lfloor U/M \rfloor$
$U = qM + Z$ $(0 \leq Z < M)$, where Z is call remainder and above equation can be represented as $Z = UmodM$.

2: Basic modular multiplication: $X \times Y \triangleq X \cdot YmodM$
Proof $XY\ mod\ M = (X\ mod\ M \cdot Y\ mod\ M)\ mod\ M$:
$X = q_1M + Z_1,\ \ Y = q_2M + Z_2$.
for LHS:
$T = XY\ mod\ M$
$T = (q_1q_2M^2 + q_2Z_1M + q_1Z_2M + Z_1Z_2)\ mod\ M$
$T = Z_1Z_2\ mod\ M$
for RHS:
$T = (X\ mod\ M \cdot Y\ mod\ M)\ mod\ M$
$T = Z_1Z_2\ mod\ M$
Hence, LHS = RHS.
Also, we can prove that $(X + Y)\ modM = (X\ mod\ M + Y\ mod\ M)\ mod\ M$.

Now, Let us discuss about algorithm 1:
First, we define X,Y as $X = \sum_{i=0}^{n_m-1} X_i2^{im}$ and $Y = \sum_{i=0}^{n_m-1} Y_i2^{im}$ in radix $r = 2^m$, where $n_m = \lceil N/m \rceil$.
To do modular multiplication $XY\ mod\ M = X(Y_{n_m-1}r^{n_m-1} + ... + Y_0r^0)\ mod\ M$:

1. $(((Z_{n_m} + XY_{n_m-1})\ mod\ M \cdot r + XY_{n_m-2})\ mod\ M \cdot r + XY_{n_m-3})\ mod\ M \cdot r\ ...\ until\ \ + XY_0)\ mod\ M$

2. $((Z_{n_m}+XY_{n_m-1})\ mod\ M \cdot r+XY_{n_m-2})\ mod\ M = (((Z_{n_m}+XY_{n_m-1})\ mod\ M \cdot r)\ mod\ M+XY_{n_m-2}\ mod\ M)\ mod\ M$

3. due to $r = 2^m$ and $2^{N-1} \leq M$, $r\ mod\ M = r$:

4. $((Z_{n_m} + XY_{n_m-1})\ mod\ M \cdot r)\ mod\ M = ((Z_{n_m} + XY_{n_m-1})\ mod\ M \cdot r\ mod\ M)\ mod\ M = ((Z_{n_m} + XY_{n_m-1})r)\ mod\ M$

5. due to $Z_{n_m} = 0$:

$((Z_{n_m} + XY_{n_m-1})r)\ mod\ M = (XY_{n_m-1}r)\ mod\ M$

6. $(((Z_{n_m}+XY_{n_m-1})\ mod\ M \cdot r)\ mod\ M+XY_{n_m-2}\ mod\ M)\ mod\ M = (XY_{n_m-1}r\ mod\ M+XY_{n_m-2}\ mod\ M)\ mod\ M = (XY_{n_m-1}r + XY_{n_m-2})\ mod\ M$

7. Hence:

$(((Z_{n_m} + XY_{n_m-1})\ mod\ M \cdot r + XY_{n_m-2})\ mod\ M \cdot r + XY_{n_m-3})\ mod\ M \cdot r\ ...\ until\ \ + XY_0)\ mod\ M = (XY_{n_m-1}r + XY_{n_m-2})\ mod\ M \cdot r + XY_{n_m-3})\ mod\ M \cdot r\ ...\ until\ \ + XY_0)\ mod\ M$

8. Iteration from $i = n_m - 1\ to\ 0$:

$(((Z_{n_m} + XY_{n_m-1})\ mod\ M \cdot r + XY_{n_m-2})\ mod\ M \cdot r + XY_{n_m-3})\ mod\ M \cdot r\ ...\ until\ \ + XY_0)\ mod\ M = ((XY_{n_m-1}r^2 + XY_{n_m-2}r + XY_{n_m-3})\ mod\ M \cdot r\ ...\ until\ \ + XY_0)\ mod\ M$

...

$(XY_{n_m-1}r^{n_m-1} + XY_{n_m-2}r^{n_m-2} + XY_{n_m-3}r^{n_m-3}... + XY_0)\ mod\ M$

9. we can conclude that algorithm 1 can achieve correct modular multiplication $XY\ mod\ M$.

Proof Z_i lies in range [0,M):
1. $Z_i = T_i - q_i M$
2. $Z_i = T_i - \lfloor T_i/M \rfloor M$
3. due to floor function, $T_i/M - \lfloor T_i/M \rfloor \in [0, 1)$
4. we can conclude that $Z_i \in [0, M)$

Then, Let us discuss about algorithm 2:
The difference between algorithm 1 and 2 is modulo operation, which is replaced by multiplication and shift operation.
1. we modify the quotient q as $\lfloor T_i/M \rfloor = \lfloor \frac{T_i}{2^{N+\beta}} \cdot \frac{2^{N+\alpha}}{M} / 2^{\alpha-\beta} \rfloor$
2. due to M, N and α is known, we represent $\frac{2^{N+\alpha}}{M}$ as μ, also, because of hardware implementation, $\mu = \lfloor \frac{2^{N+\alpha}}{M} \rfloor$.
3. Hence, above equation can be represented as $q_i = \lfloor T_i/M \rfloor = \lfloor \frac{T_i}{2^{N+\beta}} \cdot \mu / 2^{\alpha-\beta} \rfloor$
4. However, due to hardware implementation, it is hard to represent float number, so we do a further modification:
$q_i = \lfloor T_i/M \rfloor \Rightarrow q_j = \lfloor \lfloor \frac{T_i}{2^{N+\beta}} \rfloor \cdot \mu / 2^{\alpha-\beta} \rfloor$
5. Obviously, above modification produced more error while doing modulo operation:
Let us introduce the differences $e_1, e_2\ and\ e_3$:
$e_1 = \frac{T_i}{2^{N+\beta}}$ - $\lfloor \frac{T_i}{2^{N+\beta}} \rfloor$

$e_2 = \frac{2^{N+\alpha}}{M}$ - $\lfloor \frac{2^{N+\alpha}}{M} \rfloor$

$e_3 = \frac{\lfloor \frac{T_i}{2^{N+\beta}} \rfloor \mu}{2^{\alpha-\beta}}$ - $\lfloor \frac{\lfloor \frac{T_i}{2^{N+\beta}} \rfloor \mu}{2^{\alpha-\beta}} \rfloor$

e_1,e_2,$e_3 \in [0, 1)$

6. $q_j = \frac{T_i}{M}$ - $\frac{2^{N+\beta}}{M} e_1$ - $\frac{T_i}{2^{N+\alpha}} e_2$ - e_3 + $\frac{e_1 e_2}{2^{\alpha-\beta}}$

7. $Z_i = T_i - q_j M = T_i - (\frac{T_i}{M}$ - $\frac{2^{N+\beta}}{M} e_1$ - $\frac{T_i}{2^{N+\alpha}} e_2$ - e_3 + $\frac{e_1 e_2}{2^{\alpha-\beta}})M$

$Z_i = T_i - q_j M = 2^{N+\beta} e_1 + \frac{T_i}{2^{N+\alpha}} e_2 M + e_3 M - \frac{e_1 e_2}{2^{\alpha-\beta}} M$

$Z_i/M = T_i/M - q_j = e = \frac{2^{N+\beta}}{M} e_1 + \frac{T_i}{2^{N+\alpha}} e_2 + e_3 - \frac{e_1 e_2}{2^{\alpha-\beta}}$

$e < \frac{2^{N+\beta}}{2^{N-1}} e_1 + \frac{T_i}{2^{N+\alpha}} e_2 + e_3 - \frac{e_1 e_2}{2^{\alpha-\beta}}$

First we consider about the lower bound:

due to $\frac{A}{B} \geq \lfloor \frac{A}{B} \rfloor$, $\frac{T_i}{M} - q_j \geq 0$

Then we consider about the upper bound:

We can assume $0 \leq T_i = Z_{i+1} r + XY_i < 2^{N+\gamma}$, where $\gamma \geq 0$

Hence, $e < 2^{\beta+1}e_1 + 2^{\gamma-\alpha}e_2 + e_3 - \frac{e_1 e_2}{2^{\alpha-\beta}} < 2^{\beta+1}e_1 + 2^{\gamma-\alpha}e_2 + e_3 < 2^{\beta+1} + 2^{\gamma-\alpha} + 1$

To get the minimum upper bound, it is obvious that for $\beta + 1 \leq -1$ *and* $\gamma - \alpha \leq -1$, it holds $e < 2$.

Also, we need to consider intermediate result Z_i is under control:

$Z_i = Z_{i+1}r + XY_i < 2M \cdot 2^m + M \cdot 2^m < 2^{N+\gamma}$

$3M \cdot 2^m < 3 \cdot 2^{N+m} < 2^{N+\gamma} \Rightarrow 3 \cdot 2^m < 2^\gamma$

because γ needs to be an integer, we choose $\gamma \geq m + 2$.

Finally, we can conclude that when $\alpha \geq m + 3$ *and* $\beta \leq -2$, Z_i lies in the range $[0, 2M)$

2.4 Reschedule

To get faster hardware, we reschedule the iteration:

1. $q_i = \lfloor \lfloor \frac{Z_i}{2^{N+\beta}} \rfloor \mu / 2^{\alpha-\beta} \rfloor$

2. $Z_i = Z_i - q_i M$

3. $Z_{i-1} = Z_i r + XY_{i-1}$

$\Rightarrow Z_{i-1} = -q_i M r + Z_i r + XY_{i-1}$

2.5 Booth coding

2.6 Proposed Improved Barrett Modular Multiplication with Booth Coding and Operation Rescheduling

1. $ZS^{n_m} = ZC^{n_m} = 0$

2. for $i = n_m$ to 0 do

3. $ZSH^i = \lfloor ZS^i / 2^{N-2} \rfloor$, $ZCH^i = \lfloor ZC^i / 2^{N-2} \rfloor$ (get high bit number when calculating $\frac{Z_i}{2^{N+\beta}}$)

4. $Z^i_{carry} = Carry(ZS^i[N-3:N-4], ZC^i[N-3:N-4])$ (to minimize error)

5. $ZH^i = (ZSH^i, ZCH^i, Z^i_{carry})$

6. $(qS^i, qC^i) = Compress(ZH^i \cdot u)$

7. $qSH^i = \lfloor qS^i / 2^{m+5} \rfloor, qCH^i = \lfloor qC^i / 2^{m+5} \rfloor$

8. $qH^i = qSH^i + qCH^i$

9. $q^i_{carry} = Carry(qS^i[m+4:0], qC^i[m+4:0])$

10. $q^i_{carry} = (qH^i, q^i_{carry})$

11. $(ZS^{i-1}, ZC^{i-1}) = Compress(ZS^i r + ZC^i r - q^i Mr + XY_{i-1})$

12. $ZS^{i-1}_{MSB} = ZS^{i-1}_{MSB} \oplus ZC^{i-1}_{MSB}, ZC^{i-1}_{MSB} = 0$

References