Area-Time-Efficient Montgomery Modular Multiplication

Manfred Schimmler, Viktor Bunimov, Boris Tolg
Institute for Computer Engineering and Communication Networks,
Technical University of Braunschweig, Germany
Modular Multiplication

Montgomery Multiplication:

Same Idea as Interleaved Modular Multiplication:
Reduce length of intermediate result
Avoid Comparisons with M by operating LSB-first
Montgomery Multiplication

\[
\begin{array}{c}
0111 * 1011 \\
0000 \\
0111 \\
0111 \quad P_0=1? \\
1101 \\
10100 \quad \text{div} \ 2 \\
0111 \\
10001 \quad P_0=1? \\
1101 \\
11110 \quad \text{div} \ 2 \\
0000 \\
1111 \quad P_0=1? \\
1101 \\
11100 \quad \text{div} \ 2 \\
0111 \\
10101 \quad P_0=1? \\
1101 \\
100010 \quad \text{div} \ 2 \quad >M? \\
-1101 \\
\end{array}
\]

\[
X*Y*16^{-1} \mod M = 0100
\]

\[
P = 0
\]

For \( i = 0 \) to \( n-1 \) do

- \( P = P + x_i*Y \)
- If \( P_0=1 \) then \( P = P + M \)
- \( P = P \ \text{div} \ 2 \)
- If \( P > M \) then \( P = P - M \)

Result = \( P \)
Montgomery Multiplication

Needs precomputing of $16^2 \mod M$
and a second pass through the same algorithm

$\text{Temp} = X \times Y \times 16^{-1} \mod M = 0$

$P = \text{Temp} \times 16^2 \times 16^{-1} \mod M$
$\quad = X \times Y \mod M$
Carry Save Addition

3 Operands X, Y, Z    2 Results S, C

S+2*C = X+Y+Z

X: 1 0 1 0 1 1 1 1
Y: 0 1 0 1 1 1 1 1
Z: 1 1 1 1 0 0 0 1

FA

S: 0 0 0 1 1 0 0 1
C: 1 1 1 1 1 0 1 1
Carry Save Addition

3 Operands X, Y, Z  2 Results S, C

\[ S + 2C = X + Y + Z \]

Advantage: no carry propagation
addition in O(1) time

Disadvantage: difficult to compare results
even difficult to compare to 0
Algorithm for fast Montgomery Multiplication with using Carry Save Addition

- Inputs: $X, Y, M$ with $0 \leq X, Y < M$
- Output: $P = (X \times Y \times (2^n)^{-1}) \mod M$
- $n$: number of bits in $X$
- $x_i$: $i^{th}$ bit of $X$
- $s_0$: LSB of $S$

1. $S := 0; C := 0$
2. for $i:=0$ to $k-1$ do
3.   $S,C := S + C + x_i \times Y$
4.   $S,C := S + C + s_0 \times M$
5.   $S := S \div 2; C := C \div 2$
6.   $P := S + C$
7. if $P \geq M$ then $P := P - M$
Implementation of Algorithm for fast Montgomery Multiplication
Algorithm for new Montgomery Multiplication with using Carry Save Addition

• Inputs: $X, Y, M$ with $0 \leq X, Y < M$
• Output: $P = (X \times Y \times (2^n)^{-1}) \mod M$
• $n$: number of bits in $X$
• $x_i$: $i^{th}$ bit of $X$
• $s_0$: LSB of $S$, $c_0$: LSB of $C$, $y_0$: LSB of $Y$
• $R$: precomputed value of $Y+M$

1. $S := 0$; $C := 0$
2. for $i:=0$ to $k-1$ do
3. if $(s_0 = c_0)$ and not $x_0$ then $l := 0$;
4. if $(s_0 \neq c_0)$ and not $x_0$ then $l := M$;
5. if not$(s_0 \oplus c_0 \oplus y_0)$ and $x_0$ then $l := Y$;
6. if $(s_0 \oplus c_0 \oplus y_0)$ and $x_0$ then $l := R$;
7. $S, C := S + C + l$;
8. $S := S \div 2$; $C := C \div 2$;
9. $P := S + C$
10. if $P \geq M$ then $P := P - M$;
Implementation of Algorithm for new Montgomery Multiplication

\[ \text{CSA} \]

\[ \text{Shift right} \]

\[ \text{Shift right} \]
Complexity

Montgomery fast Version:
Area: 2 n-Bit-Adders
Time: 2*n loops, each of time 2
AT: 8*n²

Montgomery new Version:
Area: 1 n-Bit-Adders
Time: n loops, each of time 1
AT: 2*n²
Conclusion

The new Method is

+ time efficient
+ area efficient
+ energy efficient
+ simple

++ superior to the methods currently used!
Future work

Improving the new method by higher-radix-techniques

Modular squaring

Modular exponentiation

Implementation of applications (like RSA)