A New Algorithm for Residue Multiplication Modulo Mersenne prime $2^{521} - 1$

Shoukat Ali and Murat Cenk

Institute of Applied Mathematics
Middle East Technical University, Ankara

September 05-07, ECC 2016

- Public-key Cryptosystem
  - Factorization
  - Discrete Logarithm
- Efficiency of Modular Multiplication
  - Special primes
  - Montgomery Multiplication and Barrett Reduction

- Suppose X, Y are two residues of the Mersenne prime modulus $p = 2^{521} - 1$ and we want to compute $Z \equiv XY \pmod{p}$. Also, we are working on 64-bit machine so, we have 9-limb i.e. $521/64 = 8.1406$. In other words,

$$X = \sum_{i=0}^{8} x_i \mathrm{t}^i, \qquad Y = \sum_{i=0}^{8} y_i \mathrm{t}^i$$

$$Z = \sum_{i=0}^{8} z_i \mathrm{t}^i$$

- Since 521 is prime so, what should be the size of the limbs?
  - As done in [1, 3, 5], one can benefit by taking the size less than the bit-size of the machine i.e. 64

- Why this prime?
  - Recommended by different standards
  - Security level above 2^{250}
- The Question: can we compute this modular multiplication efficiently than the existing algorithms?
- The techniques discussed here preclude the cost of carry propagation, shifts, and/or multiplication by small constants

The cost of applying schoolbook will be

- $81\mathrm{M} + 64\mathrm{A}_d$
- $8\mathrm{A}_d$ for reduction
- $81\mathrm{M} + 72\mathrm{A}_d$

Where A_d represents double precision addition/subtraction and in our context it is 128-bit.

For the Karatsuba 3-way we use the formula of Andre Weimerskirch and Christof Paar [9] and we used two versions of this technique. We find the cost as follows:

- Recursive: $36\mathrm{M} + 54\mathrm{A} + 93\mathrm{A}_d$
- One level+Schoolbook: $54\mathrm{M} + 18\mathrm{A} + 75\mathrm{A}_d$

We use the (optimized) Toom-3 formula of Marco Bodrato [8] and and find that the recursive version is not useful. We find the cost of one level of this formula plus schoolbook as follows:

- $45\mathrm{M} + 30\mathrm{A} + 76\mathrm{A}_d$

$$
\begin{aligned}
[&x_0y_0 + x_1y_8 + x_2y_7 + x_3y_6 + x_4y_5 + x_5y_4 + x_6y_3 + x_7y_2 + x_8y_1,\\
&x_0y_1 + x_1y_0 + x_2y_8 + x_3y_7 + x_4y_6 + x_5y_5 + x_6y_4 + x_7y_3 + x_8y_2,\\
&x_0y_2 + x_1y_1 + x_2y_0 + x_3y_8 + x_4y_7 + x_5y_6 + x_6y_5 + x_7y_4 + x_8y_3,\\
&x_0y_3 + x_1y_2 + x_2y_1 + x_3y_0 + x_4y_8 + x_5y_7 + x_6y_6 + x_7y_5 + x_8y_4,\\
&x_0y_4 + x_1y_3 + x_2y_2 + x_3y_1 + x_4y_0 + x_5y_8 + x_6y_7 + x_7y_6 + x_8y_5,\\
&x_0y_5 + x_1y_4 + x_2y_3 + x_3y_2 + x_4y_1 + x_5y_0 + x_6y_8 + x_7y_7 + x_8y_6,\\
&x_0y_6 + x_1y_5 + x_2y_4 + x_3y_3 + x_4y_2 + x_5y_1 + x_6y_0 + x_7y_8 + x_8y_7,\\
&x_0y_7 + x_1y_6 + x_2y_5 + x_3y_4 + x_4y_3 + x_5y_2 + x_6y_1 + x_7y_0 + x_8y_8,\\
&x_0y_8 + x_1y_7 + x_2y_6 + x_3y_5 + x_4y_4 + x_5y_3 + x_6y_2 + x_7y_1 + x_8y_0]
\end{aligned}
$$

Let $s = \sum_{i=0}^{8} x_i y_i$ then

$$
\begin{aligned}
&[s-(x_1-x_8)(y_1-y_8)-(x_2-x_7)(y_2-y_7)-(x_3-x_6)(y_3-y_6)-(x_4-x_5)(y_4-y_5),\\
&s-(x_1-x_0)(y_1-y_0)-(x_2-x_8)(y_2-y_8)-(x_3-x_7)(y_3-y_7)-(x_4-x_6)(y_4-y_6),\\
&s-(x_5-x_6)(y_5-y_6)-(x_2-x_0)(y_2-y_0)-(x_3-x_8)(y_3-y_8)-(x_4-x_7)(y_4-y_7),\\
&s-(x_5-x_7)(y_5-y_7)-(x_2-x_1)(y_2-y_1)-(x_3-x_0)(y_3-y_0)-(x_4-x_8)(y_4-y_8),\\
&s-(x_5-x_8)(y_5-y_8)-(x_6-x_7)(y_6-y_7)-(x_3-x_1)(y_3-y_1)-(x_4-x_0)(y_4-y_0),\\
&s-(x_5-x_0)(y_5-y_0)-(x_6-x_8)(y_6-y_8)-(x_3-x_2)(y_3-y_2)-(x_4-x_1)(y_4-y_1),\\
&s-(x_5-x_1)(y_5-y_1)-(x_6-x_0)(y_6-y_0)-(x_7-x_8)(y_7-y_8)-(x_4-x_2)(y_4-y_2),\\
&s-(x_5-x_2)(y_5-y_2)-(x_6-x_1)(y_6-y_1)-(x_7-x_0)(y_7-y_0)-(x_4-x_3)(y_4-y_3),\\
&s-(x_5-x_3)(y_5-y_3)-(x_6-x_2)(y_6-y_2)-(x_7-x_1)(y_7-y_1)-(x_8-x_0)(y_8-y_0)]
\end{aligned}
$$

Cost = $9\mathrm{M} + 9 \times 4\mathrm{M} + 9 \times 8\mathrm{A} + 8\mathrm{A}_d + 9 \times 4\mathrm{A}_d$

Cost = $45\mathrm{M} + 72\mathrm{A} + 44\mathrm{A}_d$

A Toeplitz matrix or diagonal-constant matrix is a matrix in which each descending diagonal from left to right is constant i.e. an $n \times n$ Toeplitz matrix is of the following form:

$$\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & \cdots & \cdots & a_{n-1} \\ a_n & a_0 & a_1 & \ddots & \ddots & \ddots & a_{n-2} \\ a_{n+1} & a_n & a_0 & \ddots & \ddots & \ddots & a_{n-3} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & a_n & a_0 & a_1 \\ a_{2(n-1)} & \cdots & \cdots & \cdots & a_{n+1} & a_n & a_0 \end{bmatrix}$$

- Decomposition of a Toeplitz matrix results into Toeplitz (sub)matrices
- Addition and Subtraction of Toeplitz matrices is a Toeplitz matrix
- The complexity of Toeplitz Matrix-Vector Product (TMVP) is less than quadratic
- Unlike a 2-dimensional matrix, a Toeplitz matrix can be represented by first row and first column

- We observed that residue multiplication modulo $p = 2^{521} - 1$ can be presented by TMVP
- Since $521/9 = 57.88$ and like Granger-Scott in [1] we take each limb at most 58-bit but the most significant limb 57-bit. In other words, we are interested in modulus p

From schoolbook modular multiplication we have

$$
\begin{aligned}
z_0 &= f_0g_0 + 2f_8g_1 + 2f_7g_2 + 2f_6g_3 + 2f_5g_4 + 2f_4g_5 + 2f_3g_6 + 2f_2g_7 + 2f_1g_8,\\
z_1 &= f_1g_0 + f_0g_1 + 2f_8g_2 + 2f_7g_3 + 2f_6g_4 + 2f_5g_5 + 2f_4g_6 + 2f_3g_7 + 2f_2g_8,\\
z_2 &= f_2g_0 + f_1g_1 + f_0g_2 + 2f_8g_3 + 2f_7g_4 + 2f_6g_5 + 2f_5g_6 + 2f_4g_7 + 2f_3g_8,\\
z_3 &= f_3g_0 + f_2g_1 + f_1g_2 + f_0g_3 + 2f_8g_4 + 2f_7g_5 + 2f_6g_6 + 2f_5g_7 + 2f_4g_8,\\
z_4 &= f_4g_0 + f_3g_1 + f_2g_2 + f_1g_3 + f_0g_4 + 2f_8g_5 + 2f_7g_6 + 2f_6g_7 + 2f_5g_8,\\
z_5 &= f_5g_0 + f_4g_1 + f_3g_2 + f_2g_3 + f_1g_4 + f_0g_5 + 2f_8g_6 + 2f_7g_7 + 2f_6g_8,\\
z_6 &= f_6g_0 + f_5g_1 + f_4g_2 + f_3g_3 + f_2g_4 + f_1g_5 + f_0g_6 + 2f_8g_7 + 2f_7g_8,\\
z_7 &= f_7g_0 + f_6g_1 + f_5g_2 + f_4g_3 + f_3g_4 + f_2g_5 + f_1g_6 + f_0g_7 + 2f_8g_8,\\
z_8 &= f_8g_0 + f_7g_1 + f_6g_2 + f_5g_3 + f_4g_4 + f_3g_5 + f_2g_6 + f_1g_7 + f_0g_8
\end{aligned}
$$

The above expression in matrix-vector form will be

$$\begin{bmatrix} f_0 & 2f_8 & 2f_7 & 2f_6 & 2f_5 & 2f_4 & 2f_3 & 2f_2 & 2f_1 \\ f_1 & f_0 & 2f_8 & 2f_7 & 2f_6 & 2f_5 & 2f_4 & 2f_3 & 2f_2 \\ f_2 & f_1 & f_0 & 2f_8 & 2f_7 & 2f_6 & 2f_5 & 2f_4 & 2f_3 \\ f_3 & f_2 & f_1 & f_0 & 2f_8 & 2f_7 & 2f_6 & 2f_5 & 2f_4 \\ f_4 & f_3 & f_2 & f_1 & f_0 & 2f_8 & 2f_7 & 2f_6 & 2f_5 \\ f_5 & f_4 & f_3 & f_2 & f_1 & f_0 & 2f_8 & 2f_7 & 2f_6 \\ f_6 & f_5 & f_4 & f_3 & f_2 & f_1 & f_0 & 2f_8 & 2f_7 \\ f_7 & f_6 & f_5 & f_4 & f_3 & f_2 & f_1 & f_0 & 2f_8 \\ f_8 & f_7 & f_6 & f_5 & f_4 & f_3 & f_2 & f_1 & f_0 \end{bmatrix} \times \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \end{bmatrix} \quad (1)$$

Suppose we have a 3×3 TMVP then:

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & a_0 & a_1 \\ a_4 & a_3 & a_0 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} m_3 + m_4 + m_6 \\ m_2 - m_4 + m_5 \\ m_1 - m_2 - m_3 \end{bmatrix} \tag{2}$$

where

$$m_1 = (a_4 + a_3 + a_0)b_0, \quad m_2 = a_3(b_0 - b_1), \quad m_3 = a_0(b_0 - b_2),$$
$$m_4 = a_1(b_1 - b_2), \quad m_5 = (a_0 + a_3 + a_1)b_1, \quad m_6 = (a_2 + a_0 + a_1)b_2$$

- We use (2) to evaluate (1)

- The total cost of (2) is $6\mathrm{M} + 8\mathrm{A} + 6\mathrm{A}_d$
- One can apply this formula both recursively and also in combination with schoolbook.

By applying (2) on (1) we have

$$\begin{bmatrix} A_0 & 2A_2 & 2A_1 \\ A_1 & A_0 & 2A_2 \\ A_2 & A_1 & A_0 \end{bmatrix} \times \begin{bmatrix} B_0 \\ B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} M_3 + M_4 + M_6 \\ M_2 - M_4 + M_5 \\ M_1 - M_2 - M_3 \end{bmatrix}$$

For $i = 0, 1, 2$ the sub-matrices A_i are of size 3×3 and not independent, and the vectors B_i are of size 3×1

Computing M_2

$$B_0 - B_1 = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} - \begin{bmatrix} g_3 \\ g_4 \\ g_5 \end{bmatrix} = \begin{bmatrix} g_0 - g_3 \\ g_1 - g_4 \\ g_2 - g_5 \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

$$A_1(B_0 - B_1) = \begin{bmatrix} f_3 & f_2 & f_1 \\ f_4 & & \\ f_5 & & \end{bmatrix} \times \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

- Cost of M_2 : $9\mathrm{M} + 3\mathrm{A} + 6\mathrm{A}_d$

Computing M_4

$$B_1 - B_2 = \begin{bmatrix} g_3 \\ g_4 \\ g_5 \end{bmatrix} - \begin{bmatrix} g_6 \\ g_7 \\ g_8 \end{bmatrix} = \begin{bmatrix} g_3 - g_6 \\ g_4 - g_7 \\ g_5 - g_8 \end{bmatrix} = \begin{bmatrix} U_7 \\ U_8 \\ U_9 \end{bmatrix}$$

$$2A_2(B_1 - B_2) = \begin{bmatrix} 2f_6 & 2f_5 & 2f_4 \\ 2f_7 & & \\ 2f_8 & & \end{bmatrix} \times \begin{bmatrix} U_7 \\ U_8 \\ U_9 \end{bmatrix}$$

- Cost of M_4 : 9M + 3A + 6A$_d$ + 5-shift
- $2f_8$ and $2f_7$ are stored for later use

Computing M_3

$$B_0 - B_2 = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix} - \begin{bmatrix} g_6 \\ g_7 \\ g_8 \end{bmatrix} = \begin{bmatrix} g_0 - g_6 \\ g_1 - g_7 \\ g_2 - g_8 \end{bmatrix} = \begin{bmatrix} U_4 \\ U_5 \\ U_6 \end{bmatrix}$$

$$A_0(B_0 - B_2) = \begin{bmatrix} f_0 & 2f_8 & 2f_7 \\ f_1 & & \\ f_2 & & \end{bmatrix} \times \begin{bmatrix} U_4 \\ U_5 \\ U_6 \end{bmatrix}$$

- Cost of M_3 : $9\mathrm{M} + 3\mathrm{A} + 6\mathrm{A}_d$
- $2f_8$ and $2f_7$ already computed

Computing M_1

$$A_2 + A_1 = \begin{bmatrix} f_6 & f_5 & f_4 \\ f_7 & & \\ f_8 & & \end{bmatrix} + \begin{bmatrix} f_3 & f_2 & f_1 \\ f_4 & & \\ f_5 & & \end{bmatrix} = \begin{bmatrix} f_6 + f_3 & f_5 + f_2 & f_4 + f_1 \\ f_7 + f_4 & & \\ f_8 + f_5 & & \end{bmatrix}$$

$$= \begin{bmatrix} S_1 & S_2 & S_3 \\ S_4 & & \\ S_5 & & \end{bmatrix}$$

$$(A_2 + A_1) + A_0 = \begin{bmatrix} S_1 & S_2 & S_3 \\ S_4 & & \\ S_5 & & \end{bmatrix} + \begin{bmatrix} f_0 & 2f_8 & 2f_7 \\ f_1 & & \\ f_2 & & \end{bmatrix} = \begin{bmatrix} S_6 & S_7 & S_8 \\ S_9 & & \\ S_{10} & & \end{bmatrix}$$

$$(A_2 + A_1 + A_0)B_0 = \begin{bmatrix} S_6 & S_7 & S_8 \\ S_9 & & \\ S_{10} & & \end{bmatrix} \times \begin{bmatrix} g_0 \\ g_1 \\ g_2 \end{bmatrix}$$

- Cost of M_1 : 9M + 10A + 6A$_d$
- $2f_8$ and $2f_7$ already computed

Computing M_6

$$2(A_2 + A_1) + A_0 = (A_2 + A_1 + A_0) + (A_2 + A_1)$$

$$= \begin{bmatrix} S_6 & S_7 & S_8 \\ S_9 & & \\ S_{10} & & \end{bmatrix} + \begin{bmatrix} S_1 & S_2 & S_3 \\ S_4 & & \\ S_5 & & \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{14} & & \\ S_{15} & & \end{bmatrix}$$

$$((A_2 + A_1 + A_0) + (A_2 + A_1))B_2 = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{14} & & \\ S_{15} & & \end{bmatrix} \times \begin{bmatrix} g_6 \\ g_7 \\ g_8 \end{bmatrix}$$

- cost of M_6 : 9M + 5A + 6A$_d$

Computing M_5

$$(A_0 + A_1) + 2A_2 = \begin{bmatrix} f_0 + f_3 & 2f_8 + f_2 & 2f_7 + f_1 \\ f_1 + f_4 & & \\ f_2 + f_5 & & \end{bmatrix} + \begin{bmatrix} 2f_6 & 2f_5 & 2f_4 \\ 2f_7 & & \\ 2f_8 & & \end{bmatrix}$$

$$= \begin{bmatrix} S_{16} & S_{15} & S_{14} \\ S_8 & & \\ S_7 & & \end{bmatrix}$$

$$(A_0 + A_1 + 2A_2)B_1 = \begin{bmatrix} S_{16} & S_{15} & S_{14} \\ S_8 & & \\ S_7 & & \end{bmatrix} \times \begin{bmatrix} g_3 \\ g_4 \\ g_5 \end{bmatrix}$$

- Cost of M_5 : $9\mathrm{M} + 1\mathrm{A} + 6\mathrm{A}_d$
- Only compute $S_{16} = S_6 + f_6$

$$\begin{bmatrix} M_3 + M_4 + M_6 \\ M_2 - M_4 + M_5 \\ M_1 - M_2 - M_3 \end{bmatrix}$$

- Each M_i for $i = 1, \ldots, 6$ is a 3×1 vector and the elements are double-precision
- The overall cost is $54\text{M} + 25\text{A} + 54\text{A}_d + 5$-shift

We have computed the arithmetic cost of the two versions of our algorithm

- Mixed version (Proposed Technique)
- Recursive version

- Instead of using the schoolbook one may re-apply (2) to compute M_i for $i = 1, \ldots, 6$
- Cost: $36\text{M} + 73\text{A} + 54\text{A}_d$

- Worst-case scenario
  - Single precision additions/subtractions result into 60-bit
  - Single precision multiplications result into 118-bit
  - An M_i will be at most 120-bit
  - The sum of M_i will be at most 122-bit

- Using the same carry propagation as in [1], we have $[0, 2^{59} - 1] \times [0, 2^{58} - 1]^7 \times [0, 2^{57} - 1]$ where $[0, 2^{59} - 1]$ is the most significant limb and $[0, 2^{57} - 1]$ is the least significant limb
- This limb range is also used for the squaring algorithm

Technique	Arithmetic cost
Recursive version	$36\mathrm{M} + 73\mathrm{A} + 54\mathrm{A}_d$
Granger-Scott	$45\mathrm{M} + 72\mathrm{A} + 52\mathrm{A}_d$
Mixed version	$54\mathrm{M} + 25\mathrm{A} + 54\mathrm{A}_d$

Table: Number of operations for modular multiplication precluding the carrying propagation and shift

- Intel Sandy Bridge Corei5 – 2410M
- GCC 4.8.4
- At -O3
  - Granger-Scott: ~~266~~ 164
  - Version-1: ~~355~~ 179
  - ~~Version-2: 317~~


Robert Granger and Michael Scott. Faster ECC over $\mathbb{F}_{2^{521}-1}$. Public-Key Cryptography–PKC 2015. Lecture Notes in Computer Science Volume 9020, pages 539-553, 2015.


Haining Fan and M. Anwar Hasan. A New Approach to Subquadratic Space Complexity Parallel Multipliers for Extended Binary Fields. IEEE Trans. Computers 56(2), pages 224-233, 2007.


Daniel J. Bernstein, Chitchanok Chuengsatiansup, and Tanja Lange. Curve41417: Karatsuba revisited. In Cryptographic hardware and embedded systems—CHES 2014—16th international workshop, Busan, South Korea, September 23–26, 2014, proceedings, edited by Lejla Batina, Matthew Robshaw. Lecture Notes in Computer Science 8731, Springer, pages 316–334, 2014.


Daniel J. Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In Advances in cryptology–ASIACRYPT 2007, 13th international conference on the theory and application of cryptology and information security, Kuching, Malaysia, December 2–6, 2007, proceedings, edited by Kaoru Kurosawa. Lecture Notes in Computer Science 4833, Springer, pages 29–50, 2007.


Daniel J. Bernstein. Curve25519: new Diffie-Hellman speed records. In Public key cryptography-PKC 2006, 9th international conference on theory and practice in public-key cryptography, New York, USA, April 24–26, 2006, proceedings, edited by Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, Tal Malkin. Lecture Notes in Computer Science 3958, Springer, Pages 207–228, 2006.


Jerome A. Solinas. Generalized Mersenne Numbers (GMN), Technical Report, National Security Agency, Ft. Meade, MD, USA, 1999.


Darrel Hankerson, Alfred Menezes, and Scott Vanstone. Guide to Elliptic Curve Cryptography. Springer, 2004.


Marco Bodrato. Towards Optimal Toom-Cook Multiplication for Univariate and Multivariate Polynomials in Characteristic 2 and 0. In WAIFI'07 proceedings, Madrid, Spain, June 21-22, 2007, (C.Carlet and B.Sunar, eds.) Lecture Notes in Computer Science 4547, Springer, pages 116-133.


Andre Weimerskirch and Christof Paar. Generalizations of the Karatsuba Algorithm for Efficient Implementations, 2006. https://eprint.iacr.org/2006/224.pdf


Diego F. Aranha, Paulo S. L. M. Barreto, Geovandro C. C. F. Pereira, and Jefferson Ricardini. A note on high-security general-purpose elliptic curves, 2013. https://eprint.iacr.org/2013/647


Standards for Efficient Cryptography Group. SEC 2: Recommended Elliptic Curve Domain Parameters. Version 2.0, 27 January 2010, available on http://www.secg.org/sec2-v2.pdf.


US Department of Commerce, National Institute of Standards and Technology (NIST). Federal Information Processing Standards Publication (FIPS) 186-4, Digital Signature Standard (DSS). FIPS PUB 186-4, July 2013, available on http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf