Int. J. of Advanced Networking and Applications
Volume: 01. Issue: 05, Pages: 323-326 (2010)

A Montgomery Representation of Elements in GF(2^5) for Efficient Arithmetic to Use in Elliptic Curve Cryptography

A.R.Rishivarman, Assistant Professor, Mathematics ,Dr.Pauls Engineering College,,Villupuram, TN,India.

Email: rishi_130777@yahoo.co.in

B.Parthasarathy , Professor ,Mathematics, Mailam Engineering College,Villupuram,TN,India.

Email: muta_partha@yahoo.co.in

M.Thiagarajan ,Professor ,School of Computing ,SASTRA University ,Tanjore,TN,India .

Email: m_thiyagarajan@yahoo.com

ABSTRACT

Elliptic curve calculation was not introduced to cryptography until 1985. Compared with RSA, the advantage of elliptic curve cryptography lies in its ensuring the same security while the length of key of elliptic curve cryptography is much less than RSA cryptography and its lessening operation load. In this article a change of representation for elements in GF(2^5) is proposed to use in elliptic curve cryptography. The proposed representation is useful for architectures that implement Montgomery multiplication in the finite field GF(2^5). In fact, it needs virtually no cost in terms of conversion operations from a standard multiplication into a Montgomery multiplication.

Key words : Montgomery multiplications, finite field, elliptic curve cryptography (ECC), GF(2^5), irreducible polynomial.

Date of Submission: January 04, 2010 Date of Acceptance: March 03, 2010

1. Introduction

During the last few years elliptic curve cryptography (ECC) has become very popular since, among the other benefits, it uses significantly shorter keys than traditional alternatives like RSA, and is especially attractive for mobile and wireless devices which typically have limited computational resources and bandwidth. For instance, 160-bit ECC keys provide roughly the same level of security as 1024-bit RSA keys[1]. Several standardisation bodies, including ANSI, NIST, ISO, and IEEE, have adopted during the last five years standards for EC cryptography[2].

From a mathematical point of view, ECC is based on elliptic curves defined over finite fields[3], whose points form a group[1]. The main operation in ECC is the so-called point-multiplication, that is k.P where k is an integer and P is a point on the elliptic curve. Point multiplication and associated operations used in ECC are always reduced to arithmetic operations in the underlying finite fields. Efficient implementation of field arithmetic is therefore critical to the performance of ECC. This is especially true of field multiplication.

Standardization bodies recommend using either prime fields F_p or binary extension fields **GF (2^m)** as the underlying finite field for ECC[2]. The elements of F_p are the integers 0, . . . , p-1, and arithmetic on field elements is performed modulo the prime p. Elements in a finite field **GF (2^m)**, alternatively, can be represented as binary polynomials of degree m-1[3]. When such representation is used, multiplication is performed as a conventional polynomial multiplication modulo an irreducible polynomial p(x), and addition is performed by bitwise XORing the bits of element representation. For both types of finite fields, the multiplication of elements implies a reduction operation, either modulo a prime p or modulo an irreducible polynomial p(x).

A well-known algorithm for modular multiplication in F_p was proposed by Montgomery[4]. With the Montgomery algorithm, the reduction operation is interleaved with the multiplication steps by addition of multiples of the modulus, and division operations are replaced with simple shifts, which are particularly suitable for implementation in hardware and in software on general purpose computers. Therefore, the Montgomery multiplication algorithm generally allows for the design of a hardware unit with shorter signal propagation. Importantly, Montgomery multiplication enables efficient and elegant design approaches such as systolic array[5] and pipeline organizations[6]. In[7], it was shown that the Montgomery method can be extended to multiplication in **GF(2^m)**,, when polynomial basis is used. Since the steps of the Montgomery multiplication algorithm for both fields are almost identical.

The Montgomery method computes $a(x).\ b(x).\ x^{-m} \bmod p(x)$ in **GF (2^m)**, (where m correspond to the bit size of the operands), and requires a pre- and post processing conversion

Volume: 01. Issue: 05, Pages: 323-326 (2010)

of field elements. In this article we propose a change of representation for elements in **GF (2^5)**, which transforms a standard multiplication into a Montgomery multiplication at virtually no cost in terms of conversion operations. Existing research and commercial solutions implementing **GF (2^5)**, Montgomery multiplication can adopt such representation without any modification.

2.Mathematical Background

2.1The Finite Field GF(2^m)

The finite field GF(2^m) is the characteristic 2 finite field containing 2^m elements. Although there is only one characteristic 2 finite field GF(2^m) for each power 2^m of 2 with $m = 1$, there are many different ways to represent the elements of GF(2^m).

Here the elements of GF(2^m) should be represented by the set of binary polynomials of degree $m - 1$ or less:

With addition and multiplication defined in terms of an irreducible binary polynomial $f(x)$ of degree m, known as the reduction polynomial, as follows:

- Addition:

  If $a=a_{m-1}x^{m-1}+\ldots\ldots+a_0$,
  $b=b_{m-1}x^{m-1}+\ldots\ldots+b_0 \in GF(2^m)$,
  then $a+b=r \in GF(2^m)$
  ,where $r=r_{m-1}x^{m-1}+\ldots\ldots+r_0$
  with $r_i \equiv a_i + b_i \pmod 2$.

- Multiplication:

  If $a=a_{m-1}x^{m-1}+\ldots\ldots+a_0$,
  $b=b_{m-1}x^{m-1}+\ldots\ldots+b_0 \in GF(2^m)$,
  ,then $a.b = s$ in $GF(2^m)$,
  where $s=s_{m-1}x^{m-1}+\ldots\ldots+s_0$ is the remainder when the polynomial ab is divided by f(x) with all coefficient arithmetic performed modulo 2.

Addition and multiplication in GF(2^m) can be calculated efficiently using standard algorithms for ordinary integer and polynomial arithmetic . In this representation of GF(2^m), the additive identity is the polynomial 1.

Again it is convenient to define subtraction and division of field elements. To do so the additive inverse (or negative) and multiplicative inverse of a field element must be described:

- Additive inverse: if $a \in$ GF(2^m), then the additive inverse (-a) of a *in* GF(2^m) is the unique solution to the equation $a+x = 0$ in GF(2^m).
- Multiplicative inverse : if $a \in$ GF(2^m), $a \neq 0$, then the multiplicative inverse a^{-1} of a in GF(2^m) is the unique solution to the equation $a.x = 1$ in GF(2^m).

Additive inverses and multiplicative inverses in GF(2^m) can be calculated effectively using the extended Euclidean algorithm. Division and subtraction are defined in terms of additive and multiplicative inverse:

a-b in GF(2^m) is $a + (-b)$ in GF(2^m) and a/b in GF(2^m) is $a.(b^{-1})$ in GF(2^m).

2.2Elliptic Curves Over GF(2^m)

Let GF(2^m) be a characteristic 2 finite field, and let a, b $\in$ GF(2^m) satisfy b ? 0 GF(2^m). Then a (non-supersingular) elliptic curve E(GF(2^m)) over GF(2^m) defined by the parameters a, b $\in$ GF(2^m) consists of the set of solutions or points P=(x,y) for x,y $\in$ GF(2^m) to the equation:

$$y^2=x.y=x^3 + a.x^2 + b \text{ in } GF(2^m)$$

together with an extra point O called the point at infinity[11,12]. (Here the only elliptic curves over GF(2^m) of interest are non-supersingular elliptic curves.)

The number of points on E(GF(2^m)) is denoted by # E(GF(2^m)). The Hasse Theorem states that:

$$2^m + 1 - 2\sqrt{2^m} = \#\ E(GF(2^m)) = 2^m + 1 + 2\sqrt{2^m}\ .$$

It is again possible to define an addition rule to add points on E as the addition rule is specified as follows:

1. Rule to add the point at infinty to itself:
   $$O + O = O$$
2. Rule to add the point at infinity to any other point:
   $(x,y) + O = O + (x,y) = (x,y)$ for all $(x,y) \in$ GF(2^m)
3. Rule to add two points with the same –coordinates when the points are either distinct or have x-coordinates 0:
   $(x,y) + (x,x + y) = O$ for all $(x,y) \in$ GF(2^m)
4. Rule to add two points with different x-coordinates: Let $(x_1,y_1) \in$ GF(2^m) and $(x_2,y_2) \in$ GF(2^m) be two points such that $x_1 \neq x_2$. Then $(x_1,y_1)+(x_2,y_2)=(x_3,y_3)$, where:

   $x_3=?^2+?+x_1+x_2+a$ in GF(2^m),
   $y_3 = ?.(x_1+x_3)+x_3 + y_1$ in GF(2^m),
   and $? \equiv \dfrac{y_1 + y_2}{x_1 + x_2}$ in GF(2^m).

5. Rule to add a point to itself (double a point): Let $(x_1,y_1) \in$ GF(2^m) be a point with $x_1 \neq 0$. Then $(x_1,y_1) + (x_1,y_1) = (x_3,y_3)$,where:

   $x_3=?^2+?+a$ in GF(2^m) , $y_3 = x^2_1+(? +1).x_3$ in GF(2^m) ,
   and $? = x_1 + \dfrac{y_1}{x_1}$ in GF(2^m).

The set of points on E(GF(2^m)) forms an abelian group under this addition rule. Notice that the addition rule can always be computed efficiently using simple field arithmetic.

Cryptographic schemes based on ECC rely on scalar multiplication of elliptic curve points. As before given an integer k and a point $P \in$ GF(2^m) , scalar multiplication is the process of adding P *to itself* k times. The result of this scalar multiplication is denoted kP.

Volume: 01. Issue: 05, Pages: 323-326 (2010)

3.Proposed representation

Let $f(x)=x^5+x^2+1$-----(1) be an irreducible polynomial over F_2 of degree 5 and let $F_2[x]/ f(x)$ be the corresponding quotient ring. $F_2[x]/ f(x)$ is isomorphic to the field **GF (2^5)** and is used for polynomial representation. We consider the reciprocal polynomial of f(x) , denoted $\overline{f(x)}$:

We have that $\overline{f(x)}$ is irreducible and $F_2[x]/ \overline{f(x)}$ is a finite field.

We define the map as:

$$y : \frac{F_2[x]}{f(x)} \to \frac{F_2[x]}{\overline{f(x)}}, a=\sum_{i=0}^{4} a_i x^i \to \bar{a}=\sum_{i=1}^{5} \overline{a_i} x^i = \left[\sum_{i=0}^{4} a_i x^{-i}\right] x^5$$

----------(2)

Definition (2) refers to field elements $a \in F_2[x]/ f(x)$ represented as polynomials of degree less than 5. y (a)= $\bar{a}$ is the element in $F_2[x]/ \overline{f(x)}$ whose polynomial representation is obtained by reversing the bits of the polynomial representation of 'a' in $F_2[x]$ /f(x) and shifting them left by one position. We call this representation *reciprocal representation*. Note that polynomial $\bar{a}$ in (2) is of degree $\leq$ 5and $\overline{a_0}=0$.

y is bijective and has the following properties:

P1. y preserves the addition, and maps the addition in $F_2[x]/f(x)$to the addition in $F_2[x]/ \overline{f(x)}$:

$$\overline{a+b}=\bar{a}+\bar{b}$$ ----------(3)

P2. For any $a \in F_2[x]/ f(x)$ we have: $y\ (a.x)=\overline{a.x}=\bar{a}.x^{-1}$

In fact, (2) refers to field elements 'a' represented as polynomials of degree less then 5. In order to have a.x written in that from in $F_2[x]/ f(x)$ we add a suitable multiple of f(x) to it:

$$a.x = a.x+a_4.\ f(x) = \sum_{i=0}^{5}(a_{i-1}+a_4\ p_i)x^i$$

$$= \sum_{i=0}^{4}(a_{i-1}+a_4\ p_i)x^i$$ ----------(4)

where $p_5=1$ and $a_{-1}=0$

Recalling that (e+e) = 0 for any $e \in F_2=\{0,1\}$, and $P_5=1$, $a_{-1}=0$, we can rewrite $y\ (a.x)=\overline{a.x}$ in $F_2[x]/ \overline{f(x)}$ as:

$$\overline{a.x} = \overline{\sum_{i=0}^{4} a_{i-1}+a_4\ p_i)x^i} = [\sum_{i=0}^{4}(a_{i-1}+a_4\ p_i)x^{-i}]x^5$$

$$= \sum_{i=0}^{4} a_{i-1}x^{-i+5} + a_4\sum_{i=0}^{4} p_i x^{-i+5}$$

$$= \sum_{i=0}^{4} a_{i-1}x^{-i+5} + (a_4 p_5 - a_4 p_5)$$

$$+a_4\sum_{i=0}^{4} p_i x^{5-i} = \sum_{j=0}^{3} a_j x^{4-j}$$

$$+a_4 p_5 + a_4\sum_{i=0}^{5} p_i x^{5-i}$$

$$= \sum_{j=0}^{4} a_j x^{(4)-j} + a_4\overline{f(x)} = \left[\sum_{j=0}^{4} a_j x^{-j}\right] x^5.x^{-1} = \bar{a}.x^{-1}$$

----------(5)

P3.It follows form P1 and P2 that the expression in the right-hand side of (2) can be generalized to $a=\sum_{i=0}^{N-1} a_i x^i$ with an arbitrary N>5. Indeed, from the definition we have $\overline{e.x^s}=[e.x^{-s}]x^5$ for any binary coefficient $e \in F_2=\{0,1\}$ and s<5,and also, based on P2:

$$\overline{e.x^s}=\overline{e.x^{s-1}.x}=\overline{e.x^{s-1}.x^{-1}}=\overline{e.x^{s-2}.x}.x^{-1}$$

$$\overline{e.x^{s-2}}x^{-2}=.....=\overline{e.x^4}.x^{-(s-4)}$$

$$e.x^{-4}.x^5.x^{-(s-4)}=[e.x^{-s}]x^5$$ ----------(6)

for any $e \in F_2$ and $s \geq 5$. y (a)= $\bar{a}$ can thus be written as:

$$\sum_{i=0}^{\overline{N-1}}\overline{a_i x^i}=\sum_{i=0}^{N-1}\overline{a_i x^i}=\left[\sum_{i=0}^{N-1} a_i x^{-i}\right]x^5$$ for any $N \geq 5$ ---------(7)

P4.y preserves the multiplication, and maps the multiplication in $F_2[x]/ f(x)$ to the Montgomery multiplication in $F_2[x]/ \overline{f(x)}$. Indeed, in $F_2[x]/ \overline{f(x)}$ we have $\bar{a}=\left[\sum_{i=0}^{4} a_i x^{-i}\right]x^5, \bar{b}=\left[\sum_{j=0}^{4} b_j x^{-j}\right]x^5$ and the Montgomery product $\bar{a}.\bar{b}.x^{-5}$ is given by

$$[\sum_{i=0}^{4} a_i x^{-i}].x^5=[\sum_{i=0}^{4} b_j x^{-j}]x^5.x^{-5}=[\sum_{i,j<5} a_i b_j x^{-(i+j)}]x^5$$

----------(8)

Property P3 ensures that the right-hand term in the above expression can be written as

$$\left[\sum_{i,j<5} a_i b_j x^{-(i+j)}\right]x^5=\overline{\sum_{i,j<5} a_i b_j x^{i+j}}=\overline{a.b}$$ ----------(9)

In fact, y is an isomorphism from $F_2[x]/f(x)$ to $F_2[x]/ \overline{f(x)}$.The equality

$$y\ (a.b)=\overline{a.b}=\bar{a}.\bar{b}.x^{-5}$$ ----------(10)

allows us to perform interchangeably the standard polynomial multiplication in $F_2[x]$ / f(x) or the Montgomery multiplication in $F_2[x]/\overline{f(x)}$.Indeed, any composition of additions and multiplications in $F_2[x]$/ f(x) can be performed, correspondingly, as a composition of additions and Montgomery multiplications in $F_2[x]/\overline{f(x)}$.

4.Conclusion

In recent years, implementation of **GF (2^m)** Montgomery multiplication has been widely addressed by the research community[8-11]. Montgomery multiplication is often the best choice as it enables efficient systolic and pipelined designs.

Reciprocal representation can be adopted for all such solutions without any modification, as it just requires the availability of Montgomery multiplication in **GF (2^5)**. Switching to the new representation is as straightforward as a simple reversal of bits of the polynomial representation. Switching back to the original form is also straightforward since, before reversing the bits, we might just need to add $\overline{f(x)}$ to the result for zeroing its least significant bit as required for $\overline{a_0}$ by (2). Removing the Montgomery extra-factor x^5 is not necessary, as opposed to previous solutions.

Such an approach may turn out to be advantageous in all contexts where hardware support for **GF (2^5),** including embedded systems and resource-constrained environments, such as smart cards. Since any composition of additions and multiplications can be performed, correspondingly, as a composition of additions and Montgomery multiplications without any conversion cost, we indeed suggest using the reciprocal form as the representation for holding **GF (2^5)** elements in all applications.

References:

[1]. Blake, I.F., Seroussi, G., and Smart, N.P.: 'Elliptic curves in cryptography' (Cambridge University Press, 1999)

[2]. National Institute of Standards and Technology (NIST): 'Digital signature standard (DSS)' (Federal Information Processing Standards Publication 186-2, February 2000)

[3]. Lidl, R., Niederreiter, H., and Rota, G.C.: 'Finite fields (encyclopedia of mathematics and its applications)' (Cambridge University Press, 1996,2nd edn.)

[4]. Montgomery, P.L.: 'Modular multiplication without trial division', Math. Comput., 1985, 44, (170), pp. 519–521

[5]. Walter, C.D.: 'An improved linear systolic array for fast modular exponentiation', IEE Proc. Comput. Digit. Tech., 2000, 145, (5), pp. 323–328

[6]. Savaş, E., Tenca, A.F., and Koç, C¸ .K.: 'A scalable and unified multiplier architecture for finite fields GF(p) and GF(2^m)' in Cryptographic Hardware and Embedded Systems, Lect. Notes Compu. Sci. (Springer-Verlag), 2000, 1965, pp. 277–292

[7]. Koç, C¸ .K., and Acar, T.: 'Montgomery multiplication in GF(2^k)', Des. Codes Cryptogr., 1998, 14, (1), pp. 57–69

[8]. Tenca, A.F., Savas, E., and Koç, C¸ .K.: 'A design framework for scalable and unified multipliers in GF(p) and GF(2m)', Int. J. Comput. Res., 2004, 13, (1), pp. 68–83

[9]. Satoh, A., and Takano, K.: 'A scalable dual-field elliptic curve cryptographic processor', IEEE Trans. Comput., 2003, 52, (4),pp. 449–460

[10]. Großscha¨adl, J., and Kamendje, G.A.: 'Low-power design of a functional unit for arithmetic in finite fields GF(p) and GF(2^m)' in Information Security Applications, Lect. Notes Comput. Sci. (Springer Verlag), 2003, 2908, pp. 227–243

[11]. O'Rourke, C., and Sunar, B.: 'Achieving NTRU with Montgomery multiplication', IEEE Trans. Comput., 2003, 52, (4), pp. 440–448

[12]. Motorola, 'MPC180LMB Security Processor Users Manual' (available at www.freescale.com)

[13]. A.Cilardo, A.Masseo and N.Mazzoeca"Representation of Elements in F_2^mEnabeling unified field arithmetic for elliptic curve cryptography", Electronics letters July 2005 Vol:41, No.14 IEEE Xplore.