Scalar Multiplication on Koblitz Curves using τ^2-NAF

Sujoy Sinha Roy[1], Chester Rebeiro[1], Debdeep Mukhopadhyay[1],
Junko Takahashi[2] and Toshinori Fukunaga[3]

[1]Dept. of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
{*sujoyetc,chester,debdeep*}*@cse.iitkgp.ernet.in*

[2]NTT Information Sharing Platform Laboratories
Nippon Telegraph and Telephone Corporation, Japan
takahashi.junko@lab.ntt.co.jp

[3]Technology Planning Department
Nippon Telegraph and Telephone Corporation, Japan
toshi.fukunaga@hco.ntt.co.jp

Abstract. The paper proposes a τ^2-NAF method for scalar multiplication on Koblitz curves, which requires asymptotically $0.215m$ point additions in $GF(2^m)$. For τ^2-NAF method, point quading operation $(a \rightarrow a^4)$ is performed instead of point squarings. The proposed method is faster than normal $\tau-$NAF method, which requires around $\frac{m}{3}$ point additions. However, like width w based $\tau-$NAF methods, there is an overhead of pre-computations in the τ^2-NAF method. For extended binary fields of small size, the τ^2-NAF based scalar multiplication requires almost same number of point additions as in width 4 $\tau-$NAF method. Though, complexity wise, τ^2-NAF based scalar multiplication and width $4-\tau-$NAF based scalar multiplication are similar, but the techniques are different.

1 Introduction

Elliptic curve cryptography (ECC) provides more security per bit compared to other security standards. Elliptic curves are of two types: random curves and Koblitz curves. In Koblitz curves [6] Frobenius endomorphism can accelerate scalar multiplication compared to random curves.

Significant research have been carried out in accelerating scalar multiplication for hardware designs [2–5, 7, 10]. In [9], $\tau-$NAF representation of the scalar was proposed to achieve fast scalar multiplication, which does point squaring instead of point doubling. In [9], width $w-\tau-$NAF algorithm was shown to be computationally more efficient compared to normal $\tau-$NAF technique due to reduced density of non-zero digits in the representation. However there is a cost of pre-computations in width $w-\tau-$NAF.

In this paper, we present a novel τ^2-NAF method for scalar multiplication on Koblitz curves. The τ^2-NAF representation has half the length of $\tau-$NAF representation of the scalar and scalar multiplication using τ^2-NAF method requires asymptotically $0.215m$ point additions for a scalar having binary expansion length m in $GF(2^m)$. Similar to width $w-\tau-$NAF method, τ^2-NAF method also has overhead of pre-computations.

The structure of the paper is: Section 2 presents brief description of Koblitz curves. Section 3 proposes τ^2-NAF of scalar. Section 4 describes scalar multiplication technique using the proposed τ^2-NAF of scalar. The final section gives conclusion.

2 Preliminaries

In this section we present an overview of the basic properties of Koblitz curves and scalar multiplication techniques using $\tau-$NAF.

2.1 Koblitz Curves

Koblitz curves, also known as anomalous binary curves, are defined over $GF(2^m)$ and have the following representation:

$$E_a : y^2 + xy = x^3 + ax^2 + 1$$

where the elliptic curve parameter $a \in \{0, 1\}$.

The group of rational points on E_a is denoted by $E_a(2^m)$ and the order of the group is denoted by $\#E_a(2^m)$. A Koblitz curve (E_a) is said to have almost-prime order if $\#E_a(2^m) = hn$, where n is a prime and co-factor $h = 4$ for $a = 0$ and $h = 2$ for $a = 1$.

The Frobenius map $\tau : E_a(2^m) \rightarrow E_a(2^m)$ is defined by, $\tau(\infty) = \infty$ and $\tau(x, y) = (x^2, y^2)$. The Frobenius map can be considered as a complex number which follows the relation, $\tau^2 + 2 = \mu\tau$, where μ is given by $\mu = (-1)^{1-a}$. Thus $\tau = (\mu + \sqrt{-7})/2$.

The norm of an element $\alpha = (a_0 + a_1\tau) \in Z[\tau]$ is the integer product of α and its complex conjugate and is given by, $N(\alpha) = a_0^2 + \mu a_0 a_1 + 2a_1^2$. The Norm function has the following properties: (i) $N(\alpha) \geq 0$ for all $\alpha \in Z[\tau]$ with equality if and only if $\alpha = 0$. (ii) $N(\tau) = 2$ and $N(\tau - 1) = h$. (iii) Norm function is multiplicative; $N(\alpha_1\alpha_2) = N(\alpha_1)N(\alpha_2)$. (iv) The Euclidean distance from α to 0 in complex plane is $\sqrt{N(\alpha)}$. The triangle Inequality takes the form $\sqrt{N(\alpha + \beta)} \leq \sqrt{N(\alpha)} + \sqrt{N(\beta)}$.

2.2 $\tau-$adic non-adjacent form (NAF)

In [9], Solinas presented a $\tau-$NAF representation of a scalar. Any element $k \in Z[\tau]$ can be written in the form $k = \sum_{i=0}^{i=l-1} u_i\tau^i$, where each $u_i \in \{0, \pm 1\}$ and no two consecutive u_i's are nonzero. The $\tau-$NAF representation of a scalar can be obtained by repeatedly dividing the scalar by τ and obtaining the remainders (u_i's) in such a way that the NAF property is maintained. An algorithm for computing $\tau-$NAF of the scalar was presented in [9]. If the length of $\tau-$NAF representation of α be $l(\alpha)$, then $l(\alpha)$ has the following bound:

$$log_2(N(\alpha)) - 0.55 < l(\alpha) < log_2(N(\alpha)) + 3.52 \quad (1)$$

where $N(\alpha)$ is the norm of any element $\alpha \in Z[\tau]$. The average number of nonzero digits in the $\tau-$NAF representation is asymptotically 1/3 of the length [8, 9].

Now, when $\alpha = k$ for some integer scalar k, then $N(k) = k^2$. Thus, length of the $\tau-$NAF of k is approximately $log_2(N(k)) = 2log_2(k)$, which is close to twice of the length of binary expansion of k. The problem of long $\tau-$NAF representation of the scalar can be avoided by performing reduction on the scalar. In [9], Solinas presented an efficient reduction algorithm for scalar multiplication in main sub group, i.e using points of near to prime order. The reduced scalar has length close to m in $GF(2^m)$.

3 Representing a scalar using τ^2-NAF

In this section, we propose a technique to represent a reduced scalar using τ^2-NAF. This reduces the length of the scalar to almost half compared to $\tau-$NAF method.

3.1 Computation of τ^2-NAF

For any element $\alpha = c_0 + c_1\tau$, the τ^2-NAF representation can be computed by repeatedly dividing the element by τ^2 and generating the remainders as members of the τ^2-NAF.

Theorem 1. *When the element is not divisible by τ^2, the remainder r should be chosen such that,*

$$\begin{aligned} 3r &= (3c_0 - 2c_1) \ (mod\ 16) \quad for\ \mu = -1 \\ r &= (c_0 + 6c_1) \ (mod\ 16) \quad for\ \mu = 1 \end{aligned}$$

For $\mu = -1$, the set of possible remainders is thus

$$R = \{\pm 1, \pm 2, \pm 3, \pm 5, \pm 6, \pm 7, 10\}.$$

The remainder cannot take values 4 and 8, as τ^2 divides 4. At each step, after the division, the absolute values of c_0 and c_1 reduces and the algorithm terminates when both c_0 and c_1 become any of the possible values of the remainders.

The algorithm for computing the τ^2-NAF is presented in Algorithm 1 for $\mu = -1$.

3.2 Length and Density of τ^2-NAF

For $\tau-$NAF, any element α is repeatedly divided by the norm *two* element τ and thus the $\tau-$adic NAF has length close to $log_2(N(\alpha))$ [9]. For τ^2-NAF, the element is divided by the norm *four* element τ^2 and thus length is close to $log_4(N(\alpha))$. Length of τ^2-NAF has the following bound

$$log_4(N(\alpha)) - 2.9 < l < log_4(N(\alpha)) + 5.1 \tag{2}$$

It can be proved that the proposed τ^2-NAF has a nonzero density of 0.43. In $GF(2^m)$, length l of τ^2-NAF is close to $\frac{m}{2}$ and thus scalar multiplication requires nearlyasymptotically $0.215m$ point additions. The normal $\tau-$NAF method requires $\frac{m}{3}$ point additions.

Algorithm 1 $\tau^2 - adic\ NAF$

Require: *Reduced scalar* $(k_0 + k_1\tau)$
Ensure: $\tau^2 - NAF\ of\ (k_0 + k_1\tau)$
begin
Set $c_0 \leftarrow k_0,\ c_1 \leftarrow k_1$
Set $S \leftarrow <>$
While $c_0 \notin R \bigcup \{0\}$ *or* $c_1 \notin R \bigcup \{0\}$
  If $c_0 \not\equiv 2c_1\ (mod\ 4)$
    set r *so that* $3r = (3c_0 - 2c_1)\ (mod\ 16)$
    set $c_0 \leftarrow c_0 - r$
  else
    set $r \leftarrow 0$
Prepend r *to* S
Set $(c_0, c_1) \leftarrow (-\frac{2c_1+c_0}{4}, -\frac{2c_1-c_0}{4})$
EndWhile
return $< c_1, c_0, S >$

4 The τ^2-adic method for Elliptic Curve Scalar Multiplication

Let for any scalar k, the Algorithm 1 gives output c_1, c_0 and $S =< u_{l-1}, \cdots, u_0 >$. Then the scalar multiplication for base point P is given by
$kP = (c_0 + c_1\tau)(\tau^2)^l P + u_{l-1}(\tau^2)^{l-1}P + \cdots + u_1(\tau^2)P + u_0 P$.
Application of τ^2 on P is equivalent to $P(x, y) \rightarrow P(x^4, y^4)$. A single squaring is required for $c_1 \neq 0$ to compute $c_1\tau(\tau^2)^l P$. For some platforms, quading is as efficient as a single squaring [1]. The right-to-left scalar multiplication algorithm for $\mu = -1$ is presented in Algorithm 2.

Algorithm 2 Scalar Multiplication using $\tau^2 - adic\ NAF$

Require: *Scalar* k, *Base Point* P
Ensure: kP
begin
Compute $(c_0, c_1) \leftarrow Reduction(k)$
Set $Q \leftarrow \mathcal{O}$
For all $r \in R$, *compute* $P_r = rP$
While $c_0 \notin R$ *or* $c_1 \notin R$
  If $c_0 \not\equiv 2c_1\ (mod\ 4)$
    set u *so that* $3u = (3c_0 - 2c_1)\ (mod\ 16)$
    set $c_0 \leftarrow c_0 - u$
    compute $Q \leftarrow Q + P_u$
  For all $r \in R$, *compute* $P_r = \tau^2 P_r$
Set $(c_0, c_1) \leftarrow (-\frac{2c_1+c_0}{4}, -\frac{2c_1-c_0}{4})$
EndWhile
$Q \leftarrow Q + (c_0 + c_1\tau)P_1$
return Q

5 Conclusion

This paper proposes scalar multiplication on Koblitz curves using τ^2-NAF method, which results in asymptotically $0.215m$ point additions. Due to reduction in number of point additions, the proposed technique is faster than normal $\tau-$NAF method of scalar multiplication.

References

1. Chester Rebeiro, Sujoy Sinha Roy, D. Sankara Reddy and Debdeep Mukhopadhyay. Revisiting the Itoh Tsujii Inversion Algorithm for FPGA Platforms. *IEEE Trans. VLSI Syst.*, preprint.
2. K. U. Järvinen, and J. Skytta. On Parallelization of High-Speed Processors for Elliptic Curve Cryptography. *IEEE Trans. VLSI Syst.*, Sept. 2008.
3. K. U. Järvinen, and J. Skytta. Fast Point Multiplication on Koblitz Curves: Parallelization Method and Implementations. *Microproc. Microsyst. Volume 33, Issue 2*, 33:106–116, 2009.
4. K. U. Järvinen, J. Forsten, and J. Skytta. FPGA Design of Self Certified Signature Verification on Koblitz Curves. *In Cryptographic Hardware and Embedded Systems, CHES 2007, volume 4727 of Lecture Notes in Comput. Sci.*, pages 256–271, 2007.
5. Kimmo U. Järvinen and Jorma O. Skytt. High-Speed Elliptic Curve Cryptography Accelerator for Koblitz Curves. *16th International Symposium on Field-Programmable Custom Computing Machines.*
6. N. Koblitz. CM Curves with Good Cryptographic Properties. *Proc. Crypto '91*, pages 279–287, 1991.
7. J. Lutz and A. Hasan. High Performance FPGA based Elliptic Curve Cryptographic Coprocessor. *In Proc. Int. Conf. Information Technology: Coding and Computing, ITCC 2004*, 2:486–492, 2004.
8. F. Morain and J. Olivos. Speeding up the Computations on an Elliptic Curve using Addition-Subtraction Chains. *Inform. Theor. Appl.*, 24:531–543, 1990.
9. J. A. Solinas. Effecient Arithmetic on Koblitz Curves. *Design, Codes and Cryptography*, 19:195–249, 2000.
10. V.S Dimitrov, K. U. Järvinen, M. J. Jacobson, W. F. Chan, and Z. Huang. FPGA Implementation of Point Multiplication on Koblitz Curves using Kleinian Integers. *In Cryptographic Hardware and Embedded Systems, CHES 2006 , volume 4249 of Lecture Notes in Comput. Sci*, pages 445–459, 2006.