Pertanika J. Sci. & Technol. 22 (2): 489 - 505 (2014)

PERTANIKA JOURNALS

SCIENCES & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

A Reduced τ-adic Naf (RTNAF) Representation for an Efficient Scalar Multiplication on Anomalous Binary Curves (ABC)

Faridah Yunos*, Kamel Ariffin Mohd Atan, Muhammad Rezal Kamel Ariffin and Mohamad Rushdan Md Said

Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ABSTRACT

Elliptic curve cryptosystems (ECC) provides better security for each bit key utilized compared to the RSA cryptosystem. For this reason, it is projected to have more practical usage than the RSA. In ECC, scalar multiplication (or point multiplication) is the dominant operation, namely, computing nP from a point P on an elliptic curve, where n is an integer defined as the point resulting from adding $P + P + ... + P$, n times. However, for practical uses, it is very important to improve the efficiency of the scalar multiplication. Solinas (1997) proposes that the τ-adic Non-Adjacent Form (τ-NAF) is one of the most efficient algorithms used to compute scalar multiplications on Anomalous Binary curves. In this paper, we give a new property (i.e., Theorem 1.2) of τ-NAF(n) representation for every length, l. This is useful for evaluating the maximum and minimum norms occurring among all length-l elements of $Z(\tau)$. We also propose a new cryptographic method by using randomization of a multiplier n to $\bar{n}$ an element of $Z(\tau)$. It is based on τ-NAF. We focused on estimating the length of RTNAF($\bar{n}$) expansion by using a new method.

Keywords: Anomalous Binary Curves (Koblitz Curves), Scalar Multiplication, τ -adic Non-Adjacent Form, Norm.

Article history:
Received: 9 April 2012
Accepted: 17 December 2012

Email addresses:
Faridah Yunos (faridahy@upm.edu.my),
Kamel Ariffin Mohd Atan (kamel@upm.edu.my),
Muhammad Rezal Kamel Ariffin (rezal@upm.edu.my),
Mohamad Rushdan Md Said (mrushdan@upm.edu.my)
*Corresponding Author

INTRODUCTION

ECC is a cryptosystem that utilizes shorter keys as compared to the RSA and it also provides the same level of security at the same time. Thus, Vanstone (2006) suggested that ECC as the preferred asymmetric cryptosystem compared to RSA. Another advantage that is conjectured is that the elliptic curve discrete

ISSN: 0128-7702

logarithm problem is believed to be harder than the integer factorization problem. Given the best known algorithms to factor integers and compute the elliptic curve logarithms, the recommended key size is 106 bits compared to 512 bits for RSA which is considered to be an equivalent strength based on 10^4 MIPS years needed to recover one key as reported by A Certicom White Paper (1998). The scalar multiplication is the main cryptographic operation in ECC which computes $Q = nP$, where a point P is multiplied by an integer n resulting in another point Q on the elliptic curve. The computational cost of (n times) is therefore expressed as the number of field operations (additions, multiplications, inversions). The discrete logarithm problem is the basis for the security of many cryptosystems including ECC. This means that given points P and Q in the group, it is computationally infeasible to obtain n (i.e., the discrete logarithm of Q to the base P), if n is sufficiently large. Point multiplication is achieved by two basic elliptic curve operations:

1. Point addition, adding two points J and K to obtain another point (i.e., $L = J + K$) and
2. Point doubling, adding a point J to itself to obtain another point L (i.e., $L = 2J$).

Fig.1 and Fig.2 give the geometrical and analytical explanations of the point addition and the point doubling, respectively (Hankerson *et al*., 2004; Coron, 1999):

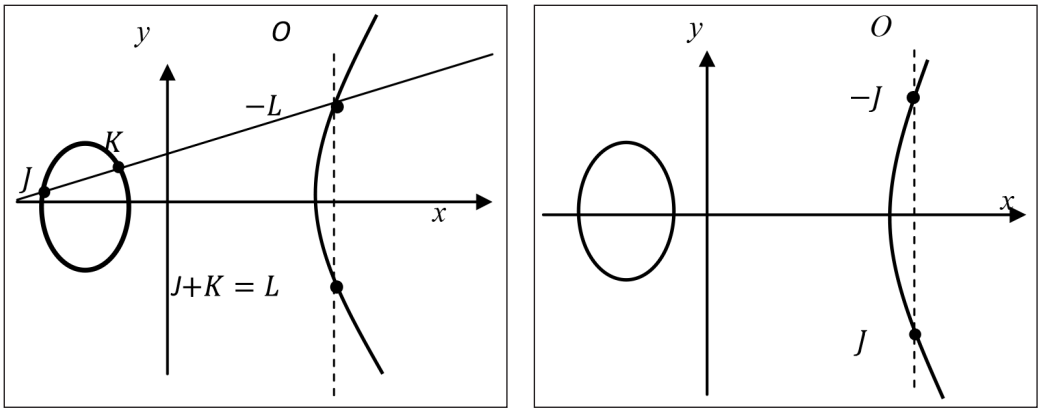


Fig.1: (Adding points in ECC): $J = (x_J, y_J)$ and $K = (x_K, y_K)$. Let $L = J + K$ where $L = (x_L, y_L)$, then $x_L = \lambda^2 - x_J - x_K$, $y_L = -y_J + \lambda(x_J - x_L)$, and $\lambda = \dfrac{y_J - y_K}{x_J - x_K}$ where λ is the slope of the line through J and K. If $K = -J$ (i.e. $K = (x_J, -y_J)$) then $J + K = O$. If $K = J$ then $J + K = 2J$ and point doubling equations are used. Also $J + K = K + J$.

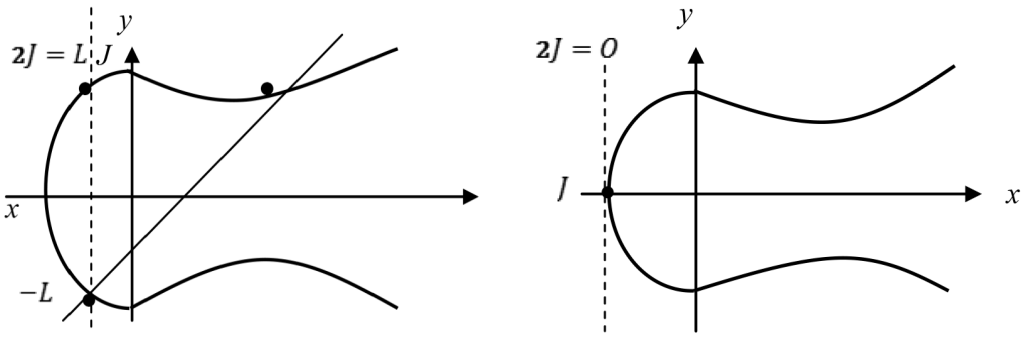


Fig.2: (Doubling points in ECC): Consider a point $J = (x_J, y_J)$ where $y_J \neq O$. Let $L = 2J$ where $L = (x_L, y_L)$. Then $x_L = \lambda^2 - 2x_J$, $y_L = -y_J + \lambda(x_J - x_L)$, and $\lambda = \frac{3x_J^2 + a}{2y_J}$ where λ is the tangent at point J and a is one of the parameters chosen with the elliptic curve.

Koblitz (1987) found that the *Koblitz curves* are a special type of curves for which the Frobenius endomorphism can be used for improving the performance of computing a scalar multiplication. The Koblitz curves are defined over F_2, as follows:

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

where $a \in \{0, 1\}$ as suggested by Koblitz (1992). The Frobenius map τ: $E_a(F_{2^m}) \rightarrow E_a(F_{2^m})$ for a point $P = (x, y)$ on $E_a(F_{2^m})$ is defined by

$$\tau(x, y) = (x^2, y^2), \qquad \tau(O) = O$$

where O is the point at infinity. It stands that $(\tau^2 + 2)P = t\tau(P)$ for all $P \in E_a(F_{2^m})$, where the trace, $t = (-1)^{1-a}$. Thus, it follows that the Frobenius map can be considered as a multiplication with complex number $\tau = \frac{t + \sqrt{-7}}{2}$ as stated earlier (Solinas, 2000).

Solinas produced an efficient procedure (i.e., Algorithm 3 in Solinas, 2000) for performing elliptic scalar multiplication and it requires about $\frac{m}{3}$ additions and no doubles. It gives at least 50% faster than any earlier version for operating on Koblitz curve, as shown in Table 1.

TABLE 1
Comparison of elliptic scalar multiplication

Method	Length of Expansion	Average Density	Average number of Elliptic Operations
Balanced expansion by Koblitz (1992)	$2m$	$\frac{3}{8}$	$3\frac{m}{4}$
Meier and Stafflebach (1993).	m	$\frac{1}{2}$	$\frac{m}{2}$
τ-NAF by Solinas (1997)	m	$\frac{1}{3}$	$\frac{m}{3}$

The computation of an average number of elliptic operations is dependent on the average density of non-zero coefficients among τ-NAF(n) representation (i.e., the number of non-zero coefficients divided by the length of expansion). Solinas (2000) estimated that the length of τ-NAF(n) is bounded by $\log_2 N(n) - 0.54626826939 < l < \log_2 N(n) + 3.5155941234$ when $l > 30$. This decision is achieved by obtaining the maximum and minimum norms occurring among all length-15 elements of Z(τ) by the direct evaluation method given by Solinas (2000, p. 213) although he does not give any detail on this method. The problem is how to get the maximum and the minimum norms of the element in Z(τ) that are more than 15 in length. In Section 1, a new property (i.e., Theorem 1.2) of τ-NAF(n) representation is given for every length, l. This is useful for evaluating the maximum and the minimum norms occurring among all the length-l elements of Z(τ) by using Equation [1.3]. In Section 2, it was observed that the property on estimation of length τ-NAF($\bar{n}$) with $l > 30$ (see Theorem 2.7). A new technique called the Reduced τ-adic Non-adjacent of Koblitz was also proposed for point multiplication for the Koblitz curves which focused on the estimation length of RTNAF.

τ- ADIC NON-ADJACENT FORM

This section begins with the meaning of a few definitions that are used in this study:

Definition 1.1

Let τ-NAF($\bar{n}$) = $\sum_{i=0}^{l-1} c_i \tau^i$ denote τ-adic Non-Adjacent Form for an integer an element of Z(τ) where l is the length of an expansion of τ-NAF($\bar{n}$), $c_i \in \{-1,0,1\}$ and $c_i c_{i+1} = 0$.

Definition 1.2

Let N: $Z(\tau) \rightarrow Z$ denote the norm function. If $x + y\tau$ is an element of Z(τ) then the norm is $x^2 + txy + 2y^2$.

It is well known that an integer of Z(τ) can be converted to τ-NAF form through Algorithm 1 in Solinas (2000). His algorithm is follows:

Algorithm 1.1 (τ-adic NAF)

Input : integers g_0, g_1
Output : τ-NAF
Computation:
Set $h_0 \leftarrow g_0$, $h_1 \leftarrow g_1$
Set $S \leftarrow \langle\ \rangle$
While $h_0 \neq 0$ or $h_1 \neq 0$
If h_0 odd
then
set $u \leftarrow 2 - (h_0 - 2h_1 \bmod 4)$
set $h_0 \leftarrow h_0 - u$

else

set $u \leftarrow 0$

Prepend u to S

Set $(h_0, h_1) \leftarrow \left(h_1 + \frac{th_0}{2}, -\frac{h_0}{2}\right)$

EndWhile

Output S

A question arises on the converse of the process. In this study, $\sum_{i=0}^{l-1} c_i \tau^i$ is transformed into an integer form with the length l in $Z(\tau)$. Thus, it is easy to get the norm of $\sum_{i=0}^{l-1} c_i \tau^i$. Furthermore, using this transformation, the maximum and minimum norms can be determined. First, the following theorem that gives an expansion of τ^i found in $\sum_{i=0}^{l-1} c_i \tau^i$ is presented.

Theorem 1.1

If $a_0 = 0$, $b_0 = 1$, $a_i = a_{i-1} + b_{i-1}$ and $b_i = -2a_{i-1}$ then

$$\tau^i = b_i t^i + a_i t^{i+1} \tau \qquad [1.1]$$

for $i > 0$.

Proof. We will give a proof by induction. If $i = 1$ then

$$\begin{aligned} \tau^1 &= b_1 t + a_1 t_2 \tau \\ &= -2a_0 t + (a_0 + b_0)\tau \\ &= \tau. \end{aligned}$$

So the equality [1.1] is verified for $i = 1$. Now, for $i = 2$,

$$\begin{aligned} \tau^2 &= b_2 t^2 + a_2 t^3 \\ &= -2a_1 t^2 + (a_1 + b_1)\, t^3 \tau \\ &= -2(a_0 + b_0)t^2 + (a_0 + b_0 - 2a_0)\, t^3 \tau \\ &= -2t^2 + t^3 \tau \\ &= t^2 + t\tau - 2 \\ &= t\tau - 2. \end{aligned}$$

So the equality [1.1] is verified for $i = 2$.
Assume that $\tau^k = b_k t^k + a_k t^{k+1} \tau$ is true up to $i = k$ where $k \geq 1$. Let us compute τ^{k+1}.

$$\begin{aligned} \tau^{k+1} &= \tau^k \bullet \tau \\ &= (b_k t^k + a_k t^{k+1} \tau)\, \tau \\ &= b_k t^k \tau + a_k t^{k+1}\, (t\tau - 2) \\ &= (b_k t^k + a_k t^{k+2})\, \tau - 2a_k t^{k+1}. \end{aligned}$$

Since $t^k = t^{k+2}t^{-2} = t^{k+2}$, we get

$$\begin{aligned}\tau^{k+1} &= (b_k t^{k+2} + a_k t^{k+2})\,\tau - 2a_k t^{k+1} \\ &= a_{k+1}t^{k+2}\tau + b_{k+1}t^{k+1}.\end{aligned}$$

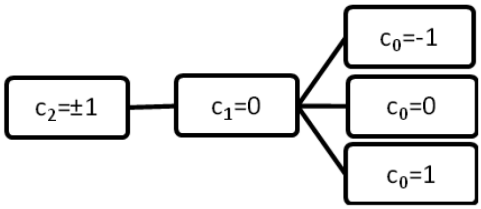
Therefore the relation [1.1] is true for all $i > 0$.

Now, let us start by making short analysis on τ-NAF that have length-3 as suggested in the following table.

TABLE 2
Combinations of c_0, c_1 and c_2 and the norm of $c_0 + c_1\tau + c_2\tau^2$

c_2	c_1	c_0	t	$r = c_0 - 2c_2$	$s = c_1 + c_2 t$	$N(r + s\tau)$
–1	0	–1	–1	1	1	2
–1	0	1	–1	3	1	8
–1	0	0	–1	2	1	4
–1	0	–1	1	1	–1	2
–1	0	1	1	3	–1	8
–1	0	0	1	2	–1	4
1	0	–1	–1	–3	–1	8
1	0	1	–1	–1	–1	2
1	0	0	–1	–2	–1	4
1	0	–1	1	–3	1	8
1	0	1	1	–1	1	2
1	0	0	1	–2	1	4

Table 2 shows all the combinations of c_0, c_1 and c_2 and the norm of $r + s\tau$ where $r + s\tau = c_0 + c_1\tau + c_2\tau^2$. From that table, we see that the maximum norm is 8 and the minimum norm is 2. There exist 12 combinations of c_0, c_1, c_2 and t to determine the maximum and minimum norm of $c_0 + c_1\tau + c_2\tau^2$. That combinations built based on the following tree diagram.



As such, there are 6 ways to arrange c_0, c_1 and c_2, while 2 ways are considered as the total number of t. Thus, there will be 12 combinations in total, as shown in Table 2. By using a similar method, all the outcomes of c_i can be obtained. Furthermore, the norms can also be obtained for every combination. Therefore, $\sum_{i=0}^{l-1} c_i \tau^i$ and the norm may be rewritten as follows:

Theorem 1.2

If $a_0 = 0$, $b_0 = 1$, $a_i = a_{i-1} + b_{i-1}$ and $b_i = -2a_{i-1}$ for $i > 0$ then

$$\sum_{i=0}^{l-1} c_i \tau^i = \sum_{i=0}^{l-1} c_i \left(b_i t^i + a_i t^{i+1} \tau\right) \qquad [1.2]$$

for $l > 0$.

Proof.

We will give a proof by induction. If $l = 1$ then $c_0 = c_0(b_0 t^0 + a_0 t^1 \tau) = c_0$.
We assume that if $l = k$ then $\sum_{i=0}^{k-1} c_i \tau^i = \sum_{i=0}^{k-1} c_i \left(b_i t^i + a_i t^{i+1} \tau\right)$ is true.
Now, if $l = k + 1$ then

$$\begin{aligned}
\sum_{i=0}^{k} c_i \tau^i &= c_0 + c_1 \tau + c_2 \tau^2 + \ldots + c_{k-1} \tau^{k-1} + c_k \tau^k \\
&= c_k \tau^k + \sum_{i=0}^{k-1} c_i \left(b_i t^i + a_i t^{i+1} \tau\right) \\
&= c_k \left(b_k t^k + a_k t^{k+1} \tau\right) + \sum_{i=0}^{k-1} c_i \left(b_i t^i + a_i t^{i+1} \tau\right) \\
&= \sum_{i=0}^{k} c_i \left(b_i t^i + a_i t^{i+1} \tau\right) \\
&= \sum_{i=0}^{(k+1)-1} c_i \left(b_i t^i + a_i t^{i+1} \tau\right)
\end{aligned}$$

Thus, [1.2] is true for $l = k + 1$ therefore it is true for all $l > 0$.

By using Definition 1.2 and Theorem 1.2, we obtain the norm of $\sum_{i=0}^{l-1} c_i \tau^i$ as follows.

$$\begin{aligned}
N\left(\sum_{i=0}^{l-1} c_i \tau^i\right) = &\left(\sum_{i=0}^{l-1} c_i b_i t^i\right)^2 + t\left(\sum_{i=0}^{l-1} c_i b_i t^i\right)\left(\sum_{i=0}^{l-1} c_i a_i t^{i+1}\right) \\
&+ 2\left(\sum_{i=0}^{l-1} c_i a_i t^{i+1}\right)^2
\end{aligned} \qquad [1.3]$$

Table 3 shows the maximum and the minimum norm of τ-NAF($\bar{n}$) occurring among all length-l elements of $Z(\tau)$ where $l = \{1,2,...,15\}$. The formula [1.3] for the norm obtained above can be made as a basis to estimate the length of τ-NAF representation, and the length of RTNAF expansion in this study. It can improve the technique to acquire the maximum and minimum norms of $\bar{n}$ by direct evaluation of all the length-15 element of $Z(\tau)$ mentioned by Solinas (2000, p. 213).

REDUCED τ-ADIC NAF

In this section, the elliptic scalar multiplication is developed on the Koblitz curve analogue of the binary method known as Reduced τ-adic Non-adjacent Form. Letting $\tau : (x,y) \rightarrow (x^2, y^2)$ the frobenius endomorphism where $\tau \in C$ will be an algebraic integer with $|\tau| > 1$. Routine

TABLE 3
The maximum and the minimum norms of τ-NAF($\bar{n}$) with l = {1,2,...,15}.

τ-NAF($\bar{n}$)	l	Maximum Norm	Minimum Norm
c_0	1	1	1
$c_0 + c_1\tau$	2	2	2
$(c_0 - 2c_2) + (c_1 + c_2t)\tau$	3	8	2
$(c_0 - 2c_2 - 2c_3t) + (c_1 + c_2t - c_3)\tau$	4	16	4
$(c_0 - 2c_2 - 2c_3t + 2c_4) + (c_1 + c_2t - c_3 - 3c_4t)\tau$	5	37	7
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t) + (c_1 + c_2t - c_3 - 3c_4t - c_5)\tau$	6	81	9
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6) + (c_1 + c_2t - c_3 - 3c_4t - c_5 - 5c_6t)\tau$	7	162	18
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t)$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7)\tau$	8	352	28
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8)$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t)\tau$	9	704	56
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8 + 6c_9t)$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t - 17c_9)\tau$	10	1421	112
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8 + 6c_9t + 34c_{10})$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t - 17c_9 - 11c_{10}t)\tau$	11	2921	197
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8 + 6c_9t + 34c_{10} + 22c_{11}t)$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t - 17c_9 - 11c_{10}t + 23c_{11})\tau$	12	5842	394
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8 + 6c_9t$ $+ 34c_{10} + 22c_{11}t - 46c_{12})$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t - 17c_9$ $- 11c_{10}t + 23c_{11} + 45c_{12}t)\tau$	13	11816	764
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8 + 6c_9t$ $+ 34c_{10} + 22c_{11}t - 46c_{12} - 90c_{13}t)$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t - 17c_9$ $- 11c_{10}t + 23c_{11} + 45c_{12}t - c_{13})\tau$	14	23662	1498
$(c_0 - 2c_2 - 2c_3t + 2c_4 + 6c_5t + 2c_6 - 10c_7t - 14c_8 + 6c_9t$ $+ 34c_{10} + 22c_{11}t - 46c_{12} - 90c_{13}t + 2c_{14})$ $+ (c_1 + c_2t - c_3 - 3c_4t - c_5 + 5c_6t + 7c_7 - 3c_8t - 17c_9$ $- 11c_{10}t + 23c_{11} + 45c_{12}t - c_{13} - 91c_{14}t)\tau$	15	47524	2996

72 in Solinas (2000) will be reused for division in $Z(\tau)$ and Algorithm 1.1 to transform an integer form in $Z(\tau)$ to τ-adic NAF expansion that was tested for its effectiveness for a long time. The division process is presented as follows:

Algorithm 2.1. (Division in $Z(\tau)$)

Input : the dividend $\gamma = u_0 + u_1\tau$ and divisor $\delta = v_0 + v_1\tau$
Output : the quotient $w = w_0 + w_1\tau$ and the remainder $z = z_0 + z_1\tau$
Computation:

$k \leftarrow u_0v_0 + tu_0v_1 + 2u_1v_1$
$l \leftarrow u_1v_0 - u_0v_1$
$N(\delta) \leftarrow v_0^2 + tv_0v_1 + 2v_1^2$

$$\lambda_0 \leftarrow \frac{k}{N(\delta)}$$

$$\lambda_1 \leftarrow \frac{l}{N(\delta)}$$

$(w_0w_1) \leftarrow$ Round (λ_0, λ_1) (Use Routine 60 in Solinas (2000) for rounding off λ_0 and λ_1)
$z_0 \leftarrow u_0 - v_0w_0 + 2v_1w_0$
$z_1 \leftarrow u_1 - v_1w_0 - v_0w_1 - tv_1w_1$
Output w_0, w_1, z_0, z_1

Our algorithm proceeds as follows:

Algorithm 2.2

(1) Consider $n \in Z$ and choose a random $\rho \in Z(\tau)$ such that $N(\rho) \leq N$ for some positive integer N;
(2) Compute the multiplier $\bar{n} \leftarrow n \mod\left(\rho\frac{\tau^m - 1}{\tau - 1}\right)$ by using Algorithm 2.1;
(3) Evaluate RTNAF of $\bar{n}$ as $\bar{n} \leftarrow n\sum_i \tau^i$ where $n_in_{i+1} = 0$ by using Algorithm 1.1;
(4) Compute Q as $Q \leftarrow \sum_i n_i\tau^i P$;
(5) Return Q.

Let $\rho = \rho_0 + \rho_1\tau$ in the element of $Z(\tau)$ and $\bar{n} \leftarrow n \mod\left(\rho\frac{\tau^m - 1}{\tau - 1}\right)$. We can find an integer ρ_0 and ρ_1 such that $N(\rho) \geq \frac{7N(n)}{4N\left(\frac{\tau^{m-1}}{\tau - 1}\right)}$ by using the following Lucas Sequence.

$$U_0 = 0,\ U_1 = 1 \qquad [2.1]$$

$$U_i = tU_{i-1} - 2U_{i-2} \quad \text{for} \quad i \geq 2 \qquad [2.2]$$

$$\tau^i = U_i\tau - 2U_{i-1} \qquad [2.3]$$

See Example 2.1 (in Appendix) in order to find ρ_0 and ρ_1.

Before multiplying the element ρ with $\frac{\tau^m - 1}{\tau - 1}$, $\frac{\tau^m - 1}{\tau - 1}$ has to be converted to $r_m + s_m\tau$ an element of $Z(\tau)$. The following theorem gives the formula of integers r_m and s_m.

Theorem 2.1

Let $\frac{\tau^m - 1}{\tau - 1} = r_m + s_m\tau$ an element of $Z(\tau)$ then $r_m = -2\sum_{i=1}^{m-2} U_i + 1$ and $s_m = \sum_{i=1}^{m-1} U_i$.

Proof.

$$\begin{aligned} r_m + s_m\tau &= \tau^{m-1} + \tau^{m-2} + \tau^{m-3} + \ldots + \tau^2 + \tau + 1 \\ &= (U_{m-1}\tau - 2U_{m-2}) + (U_{m-2}\tau - 2U_{m-3}) + (U_{m-3}\tau - 2U_{m-4}) \\ &\quad + \ldots + (U_2\tau - 2U_1) + (U_1\tau - 2U_0) + 1 \\ &= (U_{m-1} + U_{m-2} + U_{m-3} + \ldots + U_2 + U_1 + 1)\tau - 2(U_{m-2} + U_{m-3} \\ &\quad + U_{m-4} + \ldots + U_1 + U_0) + 1 \end{aligned}$$

Hence,

$$r_m = -2(U_{m-2} + U_{m-3} + U_{m-4} + \ldots + U_1) + 1 \qquad [2.4]$$

$$\equiv 1 \bmod 2 \qquad [2.5]$$

and

$$s_m = U_{m-1} + U_{m-2} + U_{m-3} + \ldots + U_2 + U_1 \qquad [2.6]$$

Now, using the formula s_m and r_m from Theorem 2.1, the following algorithm is constructed.

Algorithm 2.3: Conversion from $\rho\frac{\tau^m - 1}{\tau - 1}$ to $r + s\tau \in Z(\tau)$ by Lucas Sequence.

Input : Prime m, integer $t = (-1)^{1-a}$ for $a = 0$ or $a = 1$, nonzero elements ρ_0 and ρ_1 such that $\rho_0 + \rho_1\ \tau \in Z(\tau)$.
Output : $r + s\tau \in Z(\tau)$

(1) $U_0 \leftarrow 0,\ U_1 \leftarrow 1,\ s_0 \leftarrow 0,\ s_1 \leftarrow 1.$

(2) For i from 2 to m do $U_i \leftarrow tU_{i-1} - 2U_{i-2}$

(3) $s_m \leftarrow 1 + \sum_{i=2}^{m-1} U_i$

(4) $r_m \leftarrow -1 - 2\sum_{i=2}^{m-2} U_i$

(5) $s \leftarrow \rho_0 s_m + \rho_1 r_m + \rho_1 s_m t$

(6) $r_m \leftarrow \rho_0 r_m - 2\rho_1 s_m$

(7) Return (r, s)

The following theorem shows that the s_m in Step (3) Algorithm 2.3 is in mod 2 for $m \geq 4$.

Theorem 2.2

If $s_m = U_{m-1} + U_{m-2} + U_{m-3} + ... + U_2 + U_1$ then
$s_m \equiv$ for $t^{m-2} + t^{m-3} + t^{m-4} + ... + t^2 + t + 1 \bmod 2$ $m \geq 4$.

Proof.

$$\begin{aligned}
s_m &= U_{m-1} + U_{m-2} + U_{m-3} + ... + U_3 + U_2 + U_1 \\
&= (U_{m-2}t - 2U_{m-3}) + (U_{m-3}t - 2U_{m-4}) + (U_{m-4}t - 2U_{m-5}) + ... + (U_2t - 2U_1) + (U_1t - 2U_0) + U_1 \\
&= t(U_{m-2} + U_{m-3} + U_{m-4} + ... + U_2 + U_1) + 1 - 2(U_{m-3} + U_{m-4} + U_{m-5} + ... + U_1) \\
&= t((U_{m-3}t - 2U_{m-4}) + (U_{m-4}t - 2U_{m-5}) + (U_{m-5}t - 2U_{m-6}) + ... + U_1t + U_1) + 1 - 2(U_{m-3} + \\
&\quad U_{m-4} + U_{m-5} + ... + U_1) \\
&= t^2(U_{m-3} + U_{m-4} + U_{m-5} + ... + U_2 + U_1) + t + 1 - 2(U_{m-3} + 2U_{m-4} + 2U_{m-5} + ... + 2U_1) \\
&= t^2((U_{m-4}t - 2U_{m-5}) + (U_{m-5}t - 2U_{m-6}) + (U_{m-6}t - 2U_{m-7}) + ... + U_1t + U_1) + 1 - 2(U_{m-3} + \\
&\quad 2U_{m-4} + 2U_{m-5} + ... + 2U_1) \\
&= t^3(U_{m-4} + U_{m-5} + U_{m-6} + ... + U_1) + t^2 + t + 1 - 2(U_{m-3} + 2U_{m-4} + 3U_{m-5} + ... + 3U_1) \\
&= t^{m-2}U_{m-(m-1)} + t^{m-3} + t^{m-4} + ... + t^2 + t + 1 - 2(U_{m-3} + 2U_{m-4} + 3U_{m-5} + ... + (m-3)U_1).
\end{aligned}$$

Therefore, $s_m \equiv t^{m-2} + t^{m-3} + t^{m-4} + ... + t^2 + t + 1 \bmod 2$.

We can simplify Step (4) in Algorithm 2.3 by letting $r_m = U_m$ when $m \geq 3$ and $t = 1$.

Theorem 2.3

If $r_m = -2(U_{m-2} + U_{m-3} + U_{m-4} + ... + U_1) + 1$ and U_m is defined as in equation [2.2] then $r_m = U_m$ for $m \geq 3$ and $t = 1$.

Proof. We prove by using mathematical induction. Take $m = 3$, we have

$$r_3 = -2U_1 + 1 = -1 \text{ and } U_3 = U_2 - 2U_1 = (U_1 - 2U_0) - 2U_1 = -1$$

Therefore, for $m = 3$, $r_3 = U_3$. So the result is true for $m = 3$.
Now, our assumption asserts that $r_k = U_k$ is true for $m = k$.
Lastly, we prove that $r_{k+1} = U_{k+1}$ is true for $m = k + 1$.
We have,

$$\begin{aligned}
r_{k+1} &= -2(U_{k-1} + U_{k-2} + U_{k-3} + ... + U_1) + 1 \\
&= -2U_{k-1} - 2\,(U_{k-2} + U_{k-3} + ... + U_1) + 1 \\
&= -2U_{k+1} + r_k \\
&= -2U_{k-1} + U_k \qquad \text{using the inductive hypothesis} \\
&= U_{k+1}.
\end{aligned}$$

Hence, the result is true for all $m \geq 3$.

We give two properties of ρ as follows.

Theorem 2.4

If $\rho = \rho_0 + \rho_1\tau \in Z(\tau)$ and ρ_0 is even then f is even such that $\rho \mid (f + e\tau)$.

Proof.
Suppose $\rho_0 = 2k$ where $k \in Z$.

$$\begin{aligned}\rho(c + d\tau) &= (2k + \rho_1\tau)(c + d\tau)\\ &= 2ck + (2dk + \rho_1 c)\tau + \rho_1 d\tau^2\\ &= 2ck + (2dk + \rho_1 c)\tau + \rho_1 d(t\tau - 2)\\ &= 2(ck - \rho_1 d) + (2dk + \rho_1 c + t\rho_1 d)\tau.\end{aligned}$$

Suppose $f = 2\,(ck - \rho_1 d)$ and $e = 2dk + \rho_1 c + t\rho_1 d$. We get f is even where $ck - \rho_1 d \in Z$.

Theorem 2.5

If $\rho = \rho_0 + \rho_1\tau \in Z(\tau)$ and ρ_0 and ρ_1 are even then f and e are even such that $\rho \mid (f + e\tau)$.

Proof. Suppose $\rho_0 = 2k_1$ and $\rho_1 = 2k_2$ where $k_1, k_2 \in Z$.

$$\begin{aligned}\rho(c + d\tau) &= (2k_1 + 2k_2\tau)(c + d\tau)\\ &= 2k_1 c + 2(dk_1 + k_2 c)\tau + 2k_2 d\tau^2\\ &= 2k_1 c + 2(dk_1 + k_2 c)\tau + 2k_2 d(t\tau - 2)\\ &= 2(ck_1 - 2k_2 d) + 2(dk_1 + k_2 c + tk_2 d)\tau.\end{aligned}$$

Let $f = 2\,(ck_1 - 2k_2 d)$ and $e = 2(dk_1 + k_2 c + tk_2 d)$. Hence, we prove that f and e are even where $ck_1 - 2k_2 d,\ dk_1 + k_2 c + tk_2 d \in Z$.

If we change f and e by r and s respectively, the norm of $r + s\tau$ *is* obviously an even as well. It is helpful to observe the number of points in modulo $r + s\tau$ and the average number of non-zero coefficients in RTNAF.

Before presenting the estimation of length-l of τ-NAF($\bar{n}$), the bounds of norm of $\bar{n}$ is given as been shown in Theorem 2 of Solinas (2000). The theorem is rewritten as follows:

Theorem 2.6

Suppose that $N_{\min}(d)$ denote the minimal norm and $N_{\max}(d)$ denote the maximal norm occurring among all length-d elements of $Z(\tau)$. Let $l > 2d$, and let $\bar{n}$ be a length-l element of $Z(\tau)$. Then

$$\left(\sqrt{N_{\min}(d)} - \frac{\sqrt{N_{\max}(d)}}{2^{\frac{d}{2}} - 1}\right)^2 \cdot 2^{l-d} < N(\bar{n}) < \frac{N_{\max}(d)}{\left(2^{\frac{d}{2}} - 1\right)^2} \cdot 2^l.$$

Combining formula [1.3] and Theorem 2.6, the main result of this section can be obtained.

Theorem 2.7

The length l of τ-NAF($\bar{n}$) is bounded by

$$\log_2 N(\bar{n}) - 0.54626826939 < l < \log_2 N(\bar{n}) + 3.5155941234$$

when $l > 30$.

Proof.

We choose $d = 15$, there are 43692 combinations of c_i and t to determine the maximum and minimum norm of $\sum_{i=0}^{14} c_i \tau^i$. The norm of all the combinations is evaluated by using formula [1.3]. That is,

$$N\sum_{i=0}^{14} c_i \tau^i = \left(\sum_{i=0}^{14} c_i b_i t^i\right)^2 + t\left(\sum_{i=0}^{14} c_i b_i t^i\right)\left(\sum_{i=0}^{14} c_i a_i t^{i+1}\right) + 2\left(\sum_{i=0}^{14} c_i a_i t^{i+1}\right)^2$$

thus, we get $N_{\max}(15) = 47324$ and $N_{\min}(15) = 2996$.

Now, using Theorem 2.6, the following can be obtained:

$$\left(\sqrt{2996} - \frac{\sqrt{47324}}{2^{\frac{15}{2}} - 1}\right)^2 \cdot 2^{l-15} < N(\bar{n}) < \frac{47324}{\left(2^{\frac{15}{2}} - 1\right)^2} \cdot 2^l$$

$$0.087438100867 \cdot 2^l < N(\bar{n}) < 1.4603035291 \cdot 2^l$$

$$\log_2 N(\bar{n}) - 0.54626826939 < l < \log_2 N(\bar{n}) + 3.5155941234 \qquad [2.7]$$

when $l > 30$.

Although the limitation of the length of τ-NAF through the above theorem is similar to Solinas's study (Solinas, 2000), the approach used by him to get the maximum and the minimum norm was different from the one used in the present study. The result of this length limitation will be implemented on estimating the length of RTNAF expansion of an integer in $Z(\tau)$ in the following discussion.

Theorem 2.8

The length $\bar{l}$ of RTNAF($\bar{n}$) satisfies $\bar{l} \leq \log_2 N(\rho) + m + a$ for $\bar{l} > 30$.

Proof.

We have $N\left(\frac{\tau^m - 1}{\tau - 1}\right) = 2^{m-2+a} + O(2^{\frac{m}{2}})$ as given on page 224 of Solinas' (2000). Then, using $N(\bar{n}) \leq \frac{4}{7} N(\rho) N\left(\frac{\tau^m - 1}{\tau - 1}\right)$ the following is obtained:

$$N(\bar{n}) \leq N(\rho)\left[\frac{2^{m+a}}{7} + O(2^{\frac{m}{2}})\right]$$

$$\log_2 N(\bar{n}) \leq \log_2 N(\rho) + \log_2\left[\frac{2^{m+a}}{7} + O(2^{\frac{m}{2}})\right]$$

$$< \log_2 N(\rho) + m + a - \log_2 7 + \log_2 O(2^{\frac{m}{2}}) \qquad [2.8]$$

From equations [2.7] and [2.8], we get,

$$\bar{l} < \log_2 N(\rho) + m + a + 0.7082392013 \qquad [2.9]$$

Since $\bar{l}$ is an integer, it follows that $\bar{l} < \log_2 N(\rho) + m + a$ for $\bar{l} > 30$.

Example 2.2 (see the Appendix) is an illustration from the above theorem.

Since the hamming weight (i.e. the number of non-zero coefficients) of a scalar representation is the product of its length and density (i.e., the average of hamming weight), our bound of will help to estimate the hamming weight of scalar based on RTNAF.

Now, we give an illustration of Algorithm 2.2. Let us choose a = 1 and m = 5 to get a random ρ = 56000 + 50000τ such that $N(\rho)$ = 10936000000. Convert a divisor $\rho\frac{\tau^m - 1}{\tau - 1} = (56000 + 50000\tau)\frac{\tau^5 - 1}{\tau - 1}$ to –262000 + 144000τ by using Algorithm 2.3 and consider n = 60000000001 as a dividend. Now, compute the multiplier $\bar{n} \leftarrow n \bmod \rho\frac{\tau^m - 1}{\tau - 1}$ by using Algorithm 2.1, we obtain that the quotient is –58855 + 130678τ and the remainder is –151999 – 6000τ. Evaluate RTNAF of $\bar{n}$ as $\bar{n} \leftarrow n\sum_i \tau^i$ where $n_i n_{i+1} = 0$ by using Algorithm 1.1, we have, RTNAF(–151999 – 6000τ) =< 1,0,0,0,0,–1,0,0,0,0,–1,0,–1,0,0,1,0, –1,0,0,1,0,–1,0,0,0,1,0,–1,0,1,0,0,–1,0,0,–1> with the length is 37. Finally, we can compute the scalar multiplication by implementing any efficient algorithm such as Algorithm 3.66 in Hankerson *et al*. (2004).

CONCLUSION

As a conclusion, the new property of τ-NAF(n) representation for every length, l, as in Theorem 1.2, is useful for evaluating the maximum and the minimum norms occurring among all the length-l elements of $Z(\tau)$. An estimation of the length of RTNAF($\bar{n}$) expansion has also been presented. Retrieval from Theorem 2.8 is important to get the average of the non-zero coefficients in the RTNAF expansion that becomes the subject of our future discussion. Therefore, we can observe the effectiveness of Algorithm 2.2 in scalar multiplication as compared to the ordinary τ-NAF.

REFERENCES

A Certicom White Paper (1998). The Elliptic Curve Cryptosystems for Smart Cards. *The Seventh in a Series of ECC White Papers*. Certicom Corp, San Matco, California. http://www.certicom.com.

Coron, J. S. (1999). Resistence Against Differential Power Analysis for ECC. In C. Koc and C. Paar (Eds.), *Cryptographic Hardware and Embeddded Systems (CHES'*99*)*, *Lecturer Notes in Computer Science* 1717 (pp. 292-302). Springer-Verlag.

Gordon, D. M. (1998). A Survey of Fast Exponentiation Methods. *Journal of Algorithms 27* (*Article no AL970913*) 129-146.

Hankerson, D., Menezes, A., & Vanstone, S. (2004). Guide to Elliptic Curve Cryptography. Springer-Verlag.

Koblitz, N. (1987). Elliptic Curve Cryptosystem. *Mathematics Computation, 48*(177), 203-209.

Koblitz, N. (1992). CM Curves with Good Cryptographic Properties. *Advance in Cryptology, Proc. Crypto'91, Lecture Notes in Computer Science* 576 (pp. 279-287). Springer-Verlag.

Meier, W., & Stafflebach, O. (1993). Efficient Multiplication on Certain Non-supersingular Elliptic Curves. *Advance in Cryptology, Proc. Crypto'*92*, Lecture Notes in Computer Science, 740* (p. 333-344). Springer-Verlag.

Miller, V. S. (1986). Use of Elliptic Curve in Cryptography. In H.C. Williams (Ed.), *Advance in Cryptology, Proc.Crypto'*85, *Lecture Notes in Computer Science* 218 (pp. 417-426). Springer-Verlag.

Solinas, J. A. (1997). An Improved Algorithm for Arithmetic on a Family of Elliptic Curves. In B. Kaliski (Ed.), *Advance in Cryptology, Proc. CRYPTO'97*, *Lecture Notes in Computer Science* 1294 (pp. 357-371). Springer-Verlag.

Solinas, J. A. (2000). Efficient Arithmetic on Koblitz Curves. *Design, Codes, and Cryptography*, *9*, 195-249. Netherlands: Kluwer Academic Publishers, Boston.

Vanstone, S. (2006). ECC Holds Key to Next Generation Cryptography. Retrieved from http://www.design-reuse.com/articles/7409/ecc-hold-key-to-next-gencryptography.html.

APPENDIX

Example 2.1

Find ρ_0 and ρ_1 for $\bar{n} \equiv 10 \bmod \left(\rho \dfrac{\tau^3 - 1}{\tau - 1}\right)$ and $t = 1$.

Solution.

Let $r_3 + s_3\tau = \dfrac{\tau^3 - 1}{\tau - 1}$. Now,

$$\begin{aligned} r_3 + s_3\tau &= \tau^2 + \tau + 1 \\ &= (U_2\tau - 2U_1) + \tau + 1 \\ &= (tU_1 - 2U_0)\tau - 2U_1 + \tau + 1 \\ &= 2\tau - 1. \end{aligned}$$

Thus, we get $r_3 = -1$ and $s_3 = 2$.

Therefore, $N\left(\dfrac{\tau^3 - 1}{\tau - 1}\right) = N(-1 + 2\tau) = (-1)^2 + 1 \cdot (-1) \cdot 2 + 2 \cdot 2^2 = 7$
and $N(10) = 100$.

Now, consider $N(\rho) \geq \dfrac{7N(10)}{4N(r_3 + s_3\tau)}$ to get the value ρ_0 and ρ_1. The inequality admits $\rho_0^2 + \rho_0\rho_1 + 2\rho_1^2 \geq 25$ a solution if the discriminant $\rho_1^2 - 4(2\rho_1^2 - 25) = 100 - 7\rho_1^2$ is a positive integer, which means that $|\rho_1| \leq \dfrac{2(25)}{\sqrt{7}}$.

For a fixed positive integer $\rho_1 \leq \dfrac{2(25)}{\sqrt{7}}$, the solutions of the inequality $\rho_0^2 + \rho_0\rho_1 + 2\rho_1^2 \geq 25$ are the integers $\rho_0 \geq \dfrac{-\rho_1 + \sqrt{100 - 7\rho_1^2}}{2}$ or $\rho_0 \leq \dfrac{-\rho_1 - \sqrt{100 - 7\rho_1^2}}{2}$.

Let us choose the fixed $\rho_1 = 3$ such that $\rho_1 \leq \dfrac{2(25)}{\sqrt{7}}$. Then, by the inequality $\rho_0 \geq \dfrac{-\rho_1 + \sqrt{100 - 7\rho_1^2}}{2}$ we get $\rho_0 \geq \dfrac{-3 + \sqrt{17}}{2}$. We can take $\rho_0 = 2$. Lastly, we find that $\rho = 2 + 3\tau$ and $N(\rho) = 28$ where clearly $N(\rho) \geq 25$.

Example 2.2

Give an estimation of $\bar{l}$ when $\rho = -6000 + 6000\tau$, $m = 5$ and $a = 1$..

Solution.

Since $a = 1$ then $t = -1$, we have

$$N(-6000 + 6000\tau) = (-6000)^2 + (-1)(-6000)(6000) + 2(6000)^2$$
$$= 144000000.$$

By equation [2.9].

$$\bar{l} < \log_2 144000000 + 5 + 1 + 0.7082392013$$
$$\bar{l} < 33.8097327721$$

Since the length $\bar{l}$ is an integer, we estimate that $30 < \bar{l} \leq 33$.