SPA Resistant Scalar Multiplication Using Pell Lucas Type Chain

Shuang-Gen Liu and Hui Zhao
(Corresponding author: Shuang-Gen Liu)

Department of Information and Communication Engineering, Xi'an University of Posts and Telecommunications
Xi'an 710121, China
(Email: liusgxupt@163.com)

(Received Dec. 13, 2017; Revised and Accepted Mar. 26, 2018; First Online Mar. 4, 2019)

Abstract

A new fast and secure elliptic curve scalar multiplication algorithm is presented. The method is to utilize the front and back ratio coefficient of Pell Lucas sequences. The outcome is a new addition chain: Pell Lucas Type Chain(PLTC), and combines the mixed coordinates which shortens the previous ones. The energy curve of PLTC algorithm is unified, and can resist simple power attacks. Based on theoretical assumption and simulation experiments, it can be obtained that the new scalar multiplication by the PLTC method is 22.7 % faster than the golden ratio addition chain.

Keywords: Golden Ratio Addition Chain; Pell Lucas Type Chain; Scalar Multiplication; Simple Power Attacks

1 Introduction

Elliptic curve cryptography was proposed independently by Koblitz [15] and Miller [18] in 1985. Compared with RSA public key cryptography and EIGamal [14] public cryptography, elliptic curve cryptography provides higher security strength. For example, a 160-bit elliptic curve public key could provide comparable security to a 1024-bit RSA public key.

Hence, the elliptic curve cryptography suits the environment when the storage is limited [5,23]. The dominant operation in elliptic curve cryptography cryptographic schemes is the scalar multiplication, which is represented as kP=P+P+...+P, where P is a point given by the elliptic curve E and k is an integer, which plays the role of secret key [3]. Scalar multiplication of any one point on elliptic curves seems to be a simple addition, and yet, in the underlying field, it involves so many of multiplications. It is of great significance to find out a new method to make the chain shorter. The elliptic curve has different computational efficiency under different coordinate system. Select a suitable coordinate is critical for the scalar multiplication optimization.

There are three main operations in the underlying of the scalar multiplication: inverse, multiplication and square. The inverse is most time-consuming. Except for affine coordinate [21], coordinates which don't need inverse operation. To increase the efficiency of operation, the project coordinate [21] is often used. At the same time the Jacobian coordinate and the five element Jacobian coordinates are also used, which both proposed by Chudnovsky. It is difficult to improve the efficiency of operation by using only one coordinate [7]. But Cohen proposed that converting between the coordinates is easy, that is the characteristic of mixed coordinates [10, 19].

The core of the security chip is Cryptography algorithm. In the processing of information, there is a risk of information leakage, such as power, electromagnetic radiation, and running time. Attacker can collect and analyze the leak information then launch offensive attacks. In 1996, Kocher proposed the Side Channel Attacks (SCA) [20], it is divided into two categories: Simple Power Analysis(SPA) [6] and Differential Power Analysis (DPA) [4]. The simple power analysis is used to analyze the energy consumed by a single password operation. Because different operations have different energy consumption. For different energy consumption, an attacker can infer the order [10]. There are usually two ways to resist SPA attack. The first way is just using one kind algorithm, such as Golden Ratio Addition Chain(GRAC) [12] and the Montgomery Power Ladder [13]. The other way is to use the regular rules in algorithm, such as Double-and-add algorithm [17].

The paper presents a new $2P+Q$ algorithm using the best Mixed coordinate, which based on properties of the pell-lucas sequence and get the PLTC. The issue is mainly addressed in five parts. Part 1 gives an introduction to elliptic curve cryptography and the derivation of the pell-lucas sequence from the Lucas sequence. Part 2 introduced the new addition chain—-Pell Lucas Type Chain(PLTC). The application of PLTC in elliptic curve cryptosystems is introduced in Part 3. Part 4 makes a comparison between the PLTC and the previous algorithms under the same coordinate, at the same time,

analyze the resist of SPA attack.

2 Background

This part explain Elliptic Curve Cryptography and Pell Lucas sequence.

2.1 Elliptic Curve Cryptography

The elliptic curve E over the field K is defined by Weierstrass equation.

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \quad (1)$$

Where $a_1, a_2, a_3, a_4, a_6 \in K$ and $\Delta \neq 0$, the Δ is discriminant of E. When the characteristic of the field K is greater than 3, the equation can be simplified to:

$$E : y^2 = x^3 + ax + b. \quad (2)$$

Where $a, b \in K$ and, $\Delta = 4a^3 + 27b^2 \neq 0$.
There are two infinite points on this curve:

$$\begin{aligned} P &= (x_1, y_1), \\ Q &= (x_2, y_2), \\ P+Q &= (x_3, y_3). \end{aligned}$$

• Point Addition ($P \neq Q$)

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1, \\ \lambda &= \frac{y_2 - y_1}{x_2 - x_1}. \end{aligned} \quad (3)$$

• Point Doubling ($P = Q$)

$$\begin{aligned} x_3 &= \lambda^2 - 2x_1, \\ y_3 &= \lambda(x_1 - x_3) - y_1, \\ \lambda &= \frac{3x_1^2 + a}{2y_1} \end{aligned} \quad (4)$$

The computation of $2P+Q$ often in two methods, compute $2P$ using the double point Equation (4), add Q using Equation (3).

Equations (3) and (4) consist of multiplication, inverse and square of large integer. The three methods are represented as *M*, *I*, *S*. Comparing with these three operations, the calculation of integer addition and large integer multiplication can be ignored. S/M is equal to 0.8. The I/M ratio is generally about 10 [10, 11]. The data show that the inverse operation is the most time-consuming. Under the affine coordinates, each cost time of $2P+Q$ is $1I+9M+2S$,*Ciet*. But in [10], the realization of point addition and double point operation in other coordinates does not need to compute the inverse operation. In this paper, discussion of the complexity of algorithm is based on the Mixed coordinate, the literature [1, 9, 10] state the operation method under the Mixed coordinate [22].

2.2 Pell Lucas Sequence

The Lucas sequence is an important result of the study by Lucas in the 19th century, now it has become an important integer sequence in the Theory of Numbers. There are some inseparable links between the Lucas sequence and the Fibonacci sequence.

Definition 1. *The Fibonacci sequence is defined as* $F_n = F_{n-1} + F_{n-2} (n \geq 2)$, *and* $F_0 = 0$, $F_1 = 1$.

Definition 2. *The Lucas sequence [2] is defined as* $L_{n+1} = L_n + L_{n-1} (n = 1, 2, \cdots)$ *and* $L_0 = 2$, $L_1 = 1$. *The general equation is*

$$\begin{aligned} L_n &= \alpha^n + \alpha^n (n \geq 0) \\ \alpha &= \frac{\sqrt{5}+1}{2} \\ \beta &= \frac{1-\sqrt{5}}{2} \end{aligned}$$

It can be seen that the Fibonacci sequence and Lucas sequence are different in beginning, but the relationship between the number is same.While the Lucas sequence is consists of two linear, so there is another way to define the Lucas sequence. Take the two integers P,Q to satisfy the equation: $\Delta = P^2 - 4Q > 0$.

So we can get the equation: $x^2 - Px + Q = 0$, the roots of equation are a, b, based on this, the Lucas sequence can also be defined as

$$\begin{aligned} U_n(P, Q) &= (a^n - b^n)/(a - b), \\ V_n(P, Q) &= (a^n + b^n). \end{aligned} \quad (5)$$

Where $n \geq 0$, so we can get

$$\begin{aligned} U_0(P, Q) &= 0 \\ U_1(P, Q) &= 1 \\ V_0(P, Q) &= 2 \\ V_1(P, Q) &= P. \end{aligned}$$

If take $(P, Q) = (1, -1)$ into U_n suquence, we can get the Fibonacci sequence:

$$\{0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377 \ldots\}.$$

If take $(P, Q) = (1, -1)$ into V_n sequence,we can get the Lucas sequence:

$$\{2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843 \ldots\}.$$

When $(P, Q) = (2, -1)$, the equation $V_n(2, -1)$ is Pell-Lucas sequence, which can be represented as follows:

$$\{2, 2, 6, 14, 34, 82, 198, 418, 1154, 2786, 6726, \ldots\}.$$

At the same time, $U_n(2, -1)$ is Pell sequence:

$$\{0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, \ldots\}.$$

The general term of Pell-Lucas and Pell sequence is

(Pell-Lucas)

$$V_n = (a^n + b^n)(a = 1 - \sqrt{2}, b = 1 + \sqrt{2}),$$
$$\lim_{n\to\infty} \frac{V_n}{V_{n+1}} = \lim_{n\to\infty} \frac{a^n - b^n}{a^{n+1} - b^{n+1}} \approx 0.414. \quad (6)$$

(Pell)

$$U_n = \frac{a^n - b^n}{a - b}(a = 1 - \sqrt{2}, b = 1 + \sqrt{2}),$$
$$\lim_{n\to\infty} \frac{U_n}{U_{n+1}} = \lim_{n\to\infty} \frac{(a^n - b^n)(a - b)}{(a^{n+1} - b^{n+1})(a - b)} \approx 0.414. \quad (7)$$

It can be seen that both of Pell-Lucas or Pell sequence satisfy the following properties:

$$L_{i+1} = L_{i-1} - 2L_i (i = 1, \ldots, n), \quad (8)$$
$$L_i = L_{i+1} \times 0.414 (i = 1, \ldots, n). \quad (9)$$

3 Pell Lucas Type Chain

Equations (8) and (9) can account for the Pell-Lucas and Pell sequence, and both of the sequence satisfy Equations (8) and (9). But it's easy to see that if one sequence is corresponds to formula $L_{i+1} = L_{i-1} - 2L_i (i = 1, \ldots, n)$, it is only going to fit the formula $L_i = L_{i+1} \times 0.414 (i = 1, \ldots, n)$ at the beginning. As the extended of sequence, the ratios of front and back are deviates from 0.414. The GRAC using GAP to determine the sequence of the gold addition chain. But select the number of GAP is a new major research problem. So we define a new sequence: Pell Lucas Type Chain(PLTC).

Definition 3. *The Pell Lucas Type Chain is a sequence satisfy the formula* $L_{i+2} = L_i - 2L_{i+1} (i = 1, 2, \ldots, n)$ *and* $L_{n+1} > L_n > 0$.

The PLTC can be applied to the scalar multiplication of Elliptic curve and can greatly shorten the length of the double-and-add chain.

PLTC is not a Standard Pell-Lucas sequence. PLTC is just a chain roughly satisfies the properties of the Pell-Lucas sequence. Applying this to the elliptic curve can get Algorithm 1.

For the facilitation of the calculation,three sets $e\{\}$, $s\{\}$ and $y\{\}$ must be used in Algorithm 1. The calculation begins with the integer number k. The first step is to obtain an integer number close to $k \times 0.414$. Then we can apply $u_{i+1} = u_{i-1} - u_i \times 2 (u_i > 1, i = 1, \ldots, l)$,base on this, there will be two situations.

A: $0 < u_{i+1} < u_i \to e_i = 1$,

B: $u_{i+1} \geq u_i or u_{i+1} \leq 0 \to e_i = 0$.

$u'_{i+1} = u_{i+1} \to u'_{i+1} = s_i \to u_{i+1} = \frac{1}{2} u_i$,
if $\text{Mod}(u_{i+1}, 2) = 1 \to \{e_{i+1} = 1, y = 1\}$;
if $\text{Mod}(u_{i+1}, 2) = 0 \to \{e_{i+1} = 1, y = 0\}$.

At the last step, the number is too small, so we have two cases for the end of the reference. One is end of e=1, another one is end of e=0. Each time we will get an s or a y. We call these two end of methods are the S type end mode and Y type end mode. Each situation is shown in case Example 1 and Example 2.

Algorithm 1 Pell Lucas-Type Addition Chain

Input: A positive integer **k**
Output: $e = \{e_1, e_2, \ldots, e_i\}$
$y = \{y_1, y_2, \ldots, y_j\}, s = \{s_1, s_2, \ldots, s_j\}$
$u_0 \leftarrow \mathbf{k}$
$e\{\}$
$u_1 \leftarrow u_0 \times 0.414$
$u_2 \leftarrow u_0 - 2u_1$
$e \leftarrow e \cup \{1\}$
$s\{\}$
$y\{\}$
while $u_i > 1$ **do**
  $u_{i+1} \leftarrow u_i \times 0.414$
  $e \leftarrow e \cup \{1\}$
  $u'_{i+2} \leftarrow u_i - 2 \times u_{i-1}$
  if $0 < u'_{i+2} < u_{i+1}$ **then**
    $e \leftarrow e \cup \{1\}$
    $u_{i+2} \leftarrow u'_{i+2}$
  end if
  if $u'_{i+2} \geq u_{i+1} or u'_{i+2} \leq 0$ **then**
    $e \leftarrow e \cup \{0\}$
    $s \leftarrow s \cup \{u'_{i+2}\}$
    $u_{i+2} \leftarrow \frac{u_{i+1}}{2}$
    if $u_{i+1} mod 2 = 1$ **then**
      $e \leftarrow e \cup \{1\}$
      $y \leftarrow y \cup \{1\}$
    end if
    if $u_{i+1} mod 2 = 0$ **then**
      $e \leftarrow e \cup \{1\}$
      $y \leftarrow y \cup \{0\}$
    end if
  end if
end while

From Example 1, we can get the three sets. But if it use this data to restore the k, three sets must be reversed and get the sets like :

$$e = \{1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1\},$$
$$s = \{-1, -8, 2614\},$$
$$y = \{0, 1, 1\}.$$

Using the same method of Example 1, from Example 2, we can get the three sets:

$$e = \{0, 1, 1, 0, 1, 1\},$$
$$s = \{0, 11\},$$
$$y = \{0\}.$$

Example 1.k=131456, three sets:$e\{\}, y\{\}, s\{\}$

(**Y type end mode**)

$u_0 = \mathbf{k} = 131456$

$e = 1 \quad u_1 = u_0 \times 0.414 = 54423$

$e = 1 \quad u_2 = u_0 - 2u_1 = 22610$

$e = 1 \quad u_3 = u_1 - 2u_2 = 9203$

$e = 1 \quad u_4 = u_2 - 2u_3 = 4204$

$e = 0 \quad u_5 = u_3 - 2u_4 = 795$

$u'_6 = u_4 - 2u_5 = 2614, since 2614 > u_5,$

set $e = 0, s = s \cup \{u'_6 = 2614\}$

$since\ u_6 = \frac{u_5}{2}, and\ u_6 mod 2 = 1,$

set $e = 1, y = y \cup \{1\}$

$e = 1 \quad u_6 = \frac{u_5}{2} = 397 \cdots 1$

$e = 1 \quad u_7 = u_6 \times 0.414 = 164$

$e = 1 \quad u_8 = u_6 - 2u_7 = 69$

$e = 1 \quad u_9 = u_7 - 2u_8 = 26$

$e = 0 \quad u_{10} = u_8 - 2u_9 = 17$

$u'_{11} = u_9 - 2u_{10} = -8, since -8 < 0,$

set $e = 0, s = s \cup \{u'_{11} = -8\}$

$since\ u_{11} = \frac{u_{10}}{2}, and\ u_{11} mod 2 = 1,$

set $e = 1, y = y \cup \{1\}$

$e = 1 \quad u_{11} = \frac{u_{10}}{2} = 8 \cdots 1$

$e = 1 \quad u_{12} = u_{11} \times 0.414 = 3$

$e = 1 \quad u_{13} = u_{11} - 2u_{12} = 2$

$u'_{14} = u_{12} - 2u_{13} = -1, since -1 < 0$

set $e = 0, s = s \cup \{u'_{14} = -1\}$

$since\ u_{14} = \frac{u_{13}}{2}, and\ u_{14} mod 2 = 0,$

set $e = 1, y = y \cup \{0\}$

$e = 1 \quad u_{14} = \frac{u_{13}}{2} = 1 \cdots 0$

$since\ u_{13} - 2u_{14} = 0, and\ 0 < 1$

END

Example 2.k=175, three sets:$e\{\}, y\{\}, s\{\}$

(**S type end mode**)

$u_0 = \mathbf{k} = 175$

$e = 1 \quad u_1 = u_0 \times 0.414 = 72$

$e = 1 \quad u_2 = u_0 - 2u_1 = 31$

$e = 0 \quad u_3 = u_1 - 2u_2 = 10$

$u'_4 = u_2 - 2u_3 = 11, since 11 > u_3,$

set $e = 0, s = s \cup \{u'_4 = 11\}$

$since\ u_4 = \frac{u_3}{2}, and\ u_3 mod 2 = 0,$

set $e = 1, y = y \cup \{0\}$

$e = 1 \quad u_4 = \frac{u_3}{2} = 5 \cdots 0$

$e = 1 \quad u_5 = u_4 \times 0.414 = 2$

$e = 0 \quad u_6 = u_4 - 2u_5 = 1$

$u'_7 = u_5 - 2u_6 = 0 < 1$

set $e = 0, s = s \cup \{u'_7 = 0\}$

END

4 Application of PLTC to Elliptic Curve Cryptosystem

In Algorithm 2, there are two assignment required for each operation, T and T_0 are intermediate values in the algorithm, the cost time of assignment operation can be ignored. the last value is not remembered when the assignment end at each time, so it has no effect on memory space.

Algorithm 2 PLTC using to elliptic curve

Input: $e = \{e_1, e_2, \ldots, e_n\}, y = \{y_1, y_2, \ldots, y_i\}, s = \{s_1, s_2, \ldots, s_j\}$
Output:kP
Main loop
$i = 1$
$j = 1$
$n = 1$
if $e_n = 0$ **then**
  $T \leftarrow P$
  $P \leftarrow 2P + s_i P$
  $T_0 \leftarrow T$
  $i++$
  $n++$
end if
if $e_n = 1$ *and* $e_{n+1} = 0$ **then**
  $T \leftarrow P$
  $P \leftarrow 2P + y_j P$
  $T_0 \leftarrow T$
  $j++$
  $n++$
end if
if $e_n = 1$ *and* $e_{n+1} \neq 0$ **then**
  $T \leftarrow P$
  $P \leftarrow 2P + T_0 P$
  $T_0 \leftarrow T$
  $n++$
end if
$Q \leftarrow P$

Hence, the output is kP=Q. In Algorithm 2 operation, no mater the bit is 1 or 0, each scalar multiplication has one addition and one doubling. The two sets s and y does not affect the rate of calculation. Because all of their operations are contained in the operation of set e. Set s and set y are the fixed sequences of PLTC. These can be demonstrated in Example 3 and Example 4.

5 Discussion

5.1 Scalar Multiplication Analysis

Randomly selected 10000 of the large integers from 160 bits. Count the same chain length, According to the statistics, up to the most were 116, 117 and 118 bits. Choose the four times statistical results can obtain the Table 1. Count the length of chains from 111 to 120 and show in graph like Figure 1.

We can see from the Table 1 and Figure 1, 117 bit is always the most. The distribution of chain length is in accordance with the gaussian distribution. So the length of PLTC-160 can be seen as 117.

Example 3. e={1,0,1,1,0,1,1,1,1,0,1,1,1,1}
s={-1,-8,2614}
y={0,1,1}

$e_1 = 1, T = P, P = 2P + y_1P, T_0 = T$
$(P = 2P)$
$e_2 = 0, T = P, P = 2P + s_1P = 3P, T_0 = T$
$(P = 3P)$
$e_3 = 1, T = P, P = 2P + T_0P = 8P, T_0 = T$
$(P = 8P)$
$e_4 = 1, T = P, P = 2P + y_2P = 17P, T_0 = T$
$(P = 17P)$
$e_5 = 0, T = P, P = 2P + s_2P = 26P, T_0 = T$
$(P = 26P)$
$e_6 = 1, T = P, P = 2P + T_0P = 69P, T_0 = T$
$(P = 69P)$
$e_7 = 1, T = P, P = 2P + T_0P = 164P, T_0 = T$
$(P = 164P)$
$e_8 = 1, T = P, P = 2P + T_0P = 397P, T_0 = T$
$(P = 397P)$
$e_9 = 1, T = P, P = 2P + y_3P = 795P, T_0 = T$
$(P = 795P)$
$e_{10} = 0, T = P, P = 2P + s_3P = 4202P, T_0 = T$
$(P = 4204P)$
$e_{11} = 1, T = P, P = 2P + T_0P = 9203P, T_0 = T$
$(P = 9203P)$
$e_{12} = 1, T = P, P = 2P + T_0P = 22610P, T_0 = T$
$(P = 22610P)$
$e_{13} = 1, T = P, P = 2P + T_0P = 54423P, T_0 = T$
$(P = 54423P)$
$e_{14} = 1, T = P, P = 2P + T_0P = 131456P, T_0 = T$
$(P = 131456P)$
Q=131456P

Example 4. e={0,1,1,0,1,1}
s={0,11}
y={0}

$e_1 = 0, T = P, P = 2P + s_1P, T_0 = T$
$(P = 2P)$
$e_2 = 1, T = P, P = 2P + T_0P = 5P, T_0 = T$
$(P = 5P)$
$e_3 = 1, T = P, P = 2P + y_1P = 10P, T_0 = T$
$(P = 10P)$
$e_4 = 0, T = P, P = 2P + s_2P = 31P, T_0 = T$
$(P = 31P)$
$e_5 = 1, T = P, P = 2P + T_0P = 26P, T_0 = T$
$(P = 72P)$
$e_6 = 1, T = P, P = 2P + T_0P = 175P, T_0 = T$
$(P = 175P)$
Q=175P

We have five different kinds of coordinate systems (A, P, J, J_c, J_m) [10] that we often used. Here we compare the different cost of doubling and addition between different coordinate system. Both computation time of the operation [17] shown in Table 2 and Table 3.

Table 1: 116, 117, 118bit of PTLC

	116bit	117bit	118bit
The first time	3198	3483	1586
The second time	1844	5391	1018
The third time	758	3963	3283
The forth time	1228	4049	3037

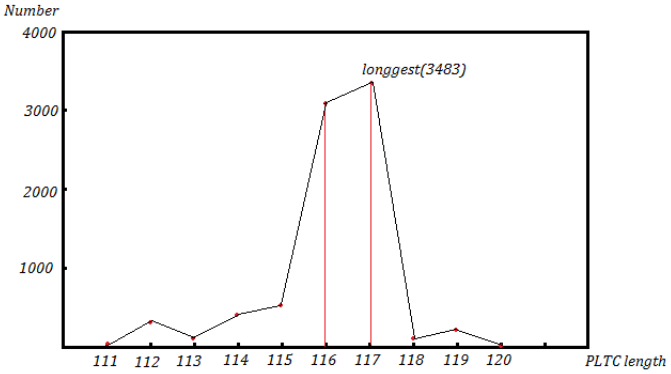

Figure 1: The number of 160-length change

Analyzing of the Table 2 and Table 3, we can obtained that the point addition operation under the J coordinates is the most time-saving operation, at the same time, the point doubling operation under the J^c coordinate is the most time-saving way. The resulting mixed coordinates are shown in Table 4.

In Algorithm 2, the calculation has one addition and one doubling each time. The length of addition is $\frac{1}{2}l$, and the length of doubling is $\frac{1}{2}l(l = 117)$. So we get the following formula.

$$\sharp[m] = l(7[M] + 7[S]).$$

The cost time of PLTC can be calculated as $1474[m]$. To effectively illustrate the advantages of PLTC algorithm. we choose to compare the number with other algorithms in the same coordinate and get Table 5.

From Table 5, we can see that under the same coordinate, PLTC is 22.7% faster than the GRAC, 7.9%, 17.2% and 29.9% faster than the 4-NAF, NAF and Double-and-add. At the same time, the reduction of chain length is considerable, and the results are shown in Table 6.

From Table 6, we can see the length of PLTC is shorter than other kind of algorithms. Even under the same length number with DFAC-160, PLTC-160 is 26.9%shorter than DFAC-160. Compared with other algorithms, PLTC is more suitable for the environments such as security chips and smart cards, which are more demanding about memory space.

Table 2: Doubling cost on different coordinates

doubling	
operation	**costs**
$2A{=}J$	$2[M]+4[S]$
$2A=J^m$	$3[M]+4[S]$
$2J^m=J$	$3[M]+4[S]$
$2A=J^c$	$3[M]+5[S]$
$2J^m$	$4[M]+4[S]$
$2J^m=J^c$	$4[M]+5[S]$
$2J$	$4[M]+6[S]$
$2J^c$	$5[M]+6[S]$
$2P$	$7[M]+5[S]$

Table 3: Addition cost on different coordinates

addition	
operation	**costs**
$P+P$	$12[M]+2[S]$
J^m+J^m	$13[M]+6[S]$
$J+A$	$8[M]+3[S]$
$J^m+A=J^m$	$9[M]+5[S]$
$J^m+A=J$	$8[M]+3[S]$
$J^c+J=J$	$11[M]+3[S]$
$J^c+J^c=J^m$	$11[M]+4[S]$
$J^c+J^c=J$	$10[M]+2[S]$
J^c+J^c	$11[M]+3[S]$
$J^c+A=J^m$	$8[M]+4[S]$
$J^c+A=J^c$	$8[M]+3[S]$
$J+A=J^m$	$9[M]+5[S]$
$A+A=J^m$	$5[M]+4[S]$
$A+A=J^c$	$5[M]+3[S]$
$J+J$	$12[M]+4[S]$
$J^c+J=J^m$	$12[M]+5[S]$
$J^m+J^c=J^m$	$12[M]+5[S]$

Table 4: Mixed coordinate

	Addition	**Doubling**
Operation	$A+A=J^c$	$2A{=}J$
Cost	$5[M]{+}3[S]$	$2[M]{+}4[S]$

Table 5: Mixed coordinate

Algorithm	Coordinate	$\sharp[m]$
Double-and-Add [17]	Mixed	2104
NAF [16, 17]	Mixed	1780
4-NAF [17]	Mixed	1600
GRAC-258 [12]	Mixed	1907
PLTC-117	Mixed	1474

Table 6: The chain length for algorithms

Algorithm	Chain Length
Fibonacci-add-add	358
Signed Fib-add-add	322
Window Fib-add-add	292
EAC-320	320
GRAC-258	258
DFAC-160	160
PLTC-160	117

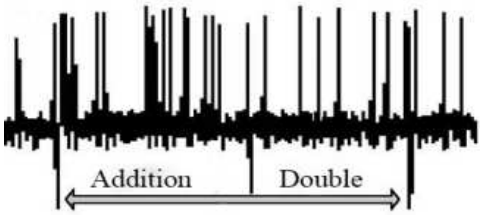

Figure 2: The power waveform of e=0

5.2 Resist SPA Analysis

The key obtained by PLTC algorithm is composed of "0" and "1", The power consumption waveforms obtained in both cases are shown in Figure 2 and Figure 3. Because the key is longer, randomly select 8 bits (1001 0001) used for PLTC coding. The power consumption waveform of a scalar multiplication in Figure 4, which collected from the power consumption analysis platform.

We can see from the three figures, that each bit has same waveform, no matter it's "0" or "1", both contains one addition and one doubling, the waveform of power is same when attacker see from outside.integrated into Figure 4, it is very hard to distinguish the energy curve, can't know the exactly number of the channel, even select a part of information. so it can resist against SPA.

6 Conclusion

This is the first study to combine Pell Lucas Type sequence with elliptic curve cryptography. With the advantages of the pell-Lucas sequence, we can improve the

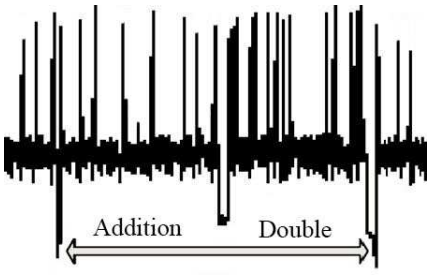

Figure 3: The power waveform of e=1

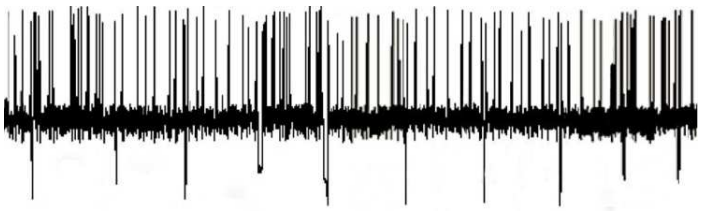

Figure 4: The power waveform of (1001 0001)

ratio between the numbers of the chain and the efficiency significantly.

For further study, we need to address the problem, although S has very little impact on the calculation and can be ignored. It accounts for about 25 %of the total chain. Therefore it will increase the burden of coding, decoding and transmission and add operations for the analysis of the password. the numbers in TABLE S can be bigger when the main chain gets longer. If we could reduce the storage space of, PLTC could be applied to elliptic curve cryptosystems more efficiently where memory is involved, such as smart card.

Acknowledgments

The support of NSFC (National Natural Science Foundation of China, No. 61272525), Jiangxi Natural Science Foundation (No. 2009GQN0094), natural science foundation research project by Shaanxi province (No. 2017JQ6010).

References

[1] D. Adachi and T. Hirata, "Combination of mixed coordinates strategy and direct computations for efficient scalar multiplications," in *IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, pp. 117–120, 2005.

[2] E. Al-Daoud, R. Mahmod, M. Rushdan, and A. Kilicman, "A new addition formula for elliptic curves over GF(2n)," *IEEE Transactions on Computers*, vol. 51, no. 8, pp. 972–975, 2002.

[3] L. M. Batten, *Public Key Cryptography: Applications and Attacks*, Wiley, 2013.

[4] T. Caddy, *Differential Power Analysis*, Springer, Boston, MA, 2005.

[5] C. C. Chang, Y. Liu and S. C. Chang, "An efficient and secure smart card based password authentication scheme," *International Journal of Network Security*, vol. 19, no. 1, pp. 1–10, 2017.

[6] T. Chen, F. Yu, K. Wu, H. Li, "Simple power analysis on elliptic curve cryptosystems and countermeasures: Practical work," *IEEE*, 2009.

[7] D. V. Chudnosky and G. V. Chudnovsky, *Sequences of Numbers Generated by Addition in Formal Groups and New Primarily and Factorization Tests*, 1986. (`https://core.ac.uk/download/pdf/82012348.pdf`)

[8] M. Ciet, M. Joye, K. Lauter, and P. L. Montgomery, "Trading inversions for multiplications in elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 39, no. 2, pp. 189–206, 2006.

[9] H. Cohen, "Analysis of the sliding window powering algorithm," *Journal of Cryptology*, vol. 18, no. 1, pp. 63–76, 2005.

[10] H. Cohen, A. Miyaji, T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," in *International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, pp. 51–65, 1998.

[11] K. Fong, D. Hankerson, J. Lopez, and A. Menezes, "Field inversion and point halving revisited," *IEEE Transactions on Computers*, vol. 53, no. 8, pp. 1047–1059, 2004.

[12] R. R. Goundar, K. I. Shiota, and M. Toyonaga, "Spa resistant scalar multiplication using golden ratio addition chain method," *Iaeng International Journal of Applied Mathematics*, vol. 38, no. 2, pp. 83–88, 2008.

[13] K. Javeed, X. Wang, "Efficient montgomery multiplier for pairing and elliptic curve based cryptography," *International Symposium on Communication Systems*, 2014. DOI: 10.1109/CSNDSP.2014.6923835

[14] J. Kar, "Id-based deniable authentication protocol based on diffie-hellman problem on elliptic curve," *International Journal of Network Security*, vol. 15, no. 5, pp. 357–364, 2013.

[15] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[16] T. C. Lin, "Algorithms on elliptic curves over fields of characteristic two with non-adjacent forms," *International Journal of Network Security*, vol. 9, no. 2, pp. 117–120, 2009.

[17] N. Meloni, "New point addition formulae for ecc applications," in *International Workshop on the Arithmetic of Finite Fields*, pp. 189–201, 2007.

[18] V. S. Miller, "Uses of elliptic curve in cryptography," *Lecture Notes in Computer Science Springer-Verlag*, vol. 218, pp. 417–426, 1986.

[19] V. Natarajan, M. Lavanya, "Improved elliptic curve arithmetic over gf(p) using different projective coordinate system," *Applied Mathematical Sciences*, vol. 9, no. 45, pp. 2235–2243, 2015.

[20] Reddy and E. Kesavulu, "Elliptic curve cryptosystems and side-channel attacks," *International Journal of Network Security*, vol. 12, no. 3, pp. 151–158, 2011.

[21] J. T. Tate, "The arithmetic of elliptic curves," *Inventiones Mathematicae*, vol. 23, pp. 179–206, 1974.

[22] X. C. Yin and H. X. Hou, "Improved sliding window scalar multiplication algorithm," *Journal of Chinese Computer Systems*, vol. 29, no. 5, pp. 863–866, 2008.

[23] D. Yong, Y. F. Hong, W. T. Wang, Y. Y. Zhou, and X. Y. Zhao, "Speeding scalar multiplication of

elliptic curve over gf(2)," *International Journal of Network Security*, vol. 11, no. 10, pp. 70–77, 2010.

Biography

Shuang-Gen Liu was born in 1979, associate professor. He graduated from Xidian University in 2008 with a major in cryptography, PhD, a member of the Chinese Institute of computer science, and a member of the Chinese code society.

Hui Zhao is a graduate student of Xi'an University of post and telecommunications. She is mainly engaged in the research of elliptic curve cryptosystem.