A More Flexible Countermeasure against Side Channel Attacks Using Window Method

Katsuyuki Okeya[1] and Tsuyoshi Takagi[2]

[1] Hitachi, Ltd., Systems Development Laboratory,
292, Yoshida-cho, Totsuka-ku, Yokohama, 244-0817, Japan
ka-okeya@sdl.hitachi.co.jp
[2] Technische Universität Darmstadt, Fachbereich Informatik,
Alexanderstr.10, D-64283 Darmstadt, Germany
ttakagi@cdc.informatik.tu-darmstadt.de

Abstract. Elliptic curve cryptosystem (ECC) is well-suited for the implementation on memory constraint environments due to its small key size. However, side channel attacks (SCA) can break the secret key of ECC on such devices, if the implementation method is not carefully considered. The scalar multiplication of ECC is particularly vulnerable to the SCA. In this paper we propose an SCA-resistant scalar multiplication method that is allowed to take any number of pre-computed points. The proposed scheme essentially intends to resist the simple power analysis (SPA), not the differential power analysis (DPA). Therefore it is different from the other schemes designed for resisting the DPA. The previous SPA-countermeasures based on window methods utilize the fixed pattern windows, so that they only take discrete table size. The optimal size is 2^{w-1} for $w = 2, 3, ...$, which was proposed by Okeya and Takagi. We play a different approach from them. The key idea is randomly (but with fixed probability) to generate two different patterns based on pre-computed points. The two distributions are indistinguishable from the view point of the SPA. The proposed probabilistic scheme provides us more flexibility for generating the pre-computed points — the designer of smart cards can freely choose the table size without restraint.

Keywords: Elliptic Curve Cryptosystem, Side Channel Attacks, Width-w NAF, Fractional window, Pre-computation Table, Smart Card, Memory Constraint

1 Introduction

We are standing to the beginning of the ubiquitous computing era. It is expected that we can accomplish lucrative applications by effectively synthesizing the ubiquitous computer with cryptography. The ubiquitous computer only has scarce computational environments, so that we have to make an effort to optimize the memory and efficiency of the cryptosystem. Elliptic curve cryptosystem (ECC) is suitable for the purpose because of its short key size [Kob87,Mil86]. However, several experimental tests show that side channel attacks (SCA) can

C.D. Walter et al. (Eds.): CHES 2003, LNCS 2779, pp. 397–410, 2003.
© Springer-Verlag Berlin Heidelberg 2003

break the ECC if the implementation on the devices is not carefully considered [Cor99,HaM02,IIT02].

The SCA tries to find a correlation between the side channel information and the operation related to the secret key. In this paper we discuss the SCA on ECC using the power analysis [KJJ99], which consists of the simple power analysis (SPA) and the differential power analysis (DPA). The SPA simply observes several power consumptions of the device, and the DPA is additionally allowed to use a statistical tool in order to guess the secret information. An SPA-resistant scheme can be converted to be a DPA-resistant one by randomizing the parameters of the underlying system (See for example [Cor99,JT01]). There are three different types of SPA-resistant scheme: (1) indistinguishable addition formula that uses one formula for both of elliptic addition and doubling [LS01,JQ01, BJ02]. (2) addition chain that always computes elliptic addition and doubling for each bit [Cor99,OS00,FGKS02,IT02,BJ02]. (3) window based addition chain with fixed pattern [Möl01a,Möl01b,Möl02a,OT03]. In this paper we deal with the third category. The optimal one in (3) is the scheme proposed by Okeya and Takagi [OT03].

We intend to propose an SPA-resistant scalar multiplication that allows us to choose any number of the pre-computed points. We try to reduce the table size of the Okeya-Takagi scheme using the fractional window method proposed by Möller [Möl02b]. The fractional window method reduces a part of pre-computed points to smaller window size. Therefore, the table length is not fixed anymore, and the corresponding addition chain has no fixed pattern. It is not obvious to construct an SPA-resistant scheme using the fractional window method. In order to overcome this bias we propose a novel approach. We generate the points with smaller window size as the probabilistic process, which are indistinguishable from the view point of the SPA. Indeed, all points in the table are classified: (*lower*) points (i.e. $uP, u < 2^{w-1}$) and (*upper*) points (i.e. $uP, u > 2^{w-1}$), where w and P is the underlying width and the base point, respectively. We control the reduction probability of (*lower*) based on that of (*upper*), namely the distribution of both (*lower*) and (*upper*) are indistinguishable against SPA. The pre-computed points for (*upper*) are randomly chosen for every scalar multiplication, and the points in class (*lower*) are randomly reduced with the above reduction probability. Thus the SPA cannot detect which point is used in each class (*upper*) and (*lower*).

In order to implement highly functional applications on memory constraint device such as smartcards, the cryptographic functions are usually required to be efficient and to use small memory. In addition, some applications are often appended to (deleted from) the smartcards, thus the memory space allowed to use for the cryptographic functions depends on such individual situations. Hence the cryptographic schemes should be optimized on such individual situations. The proposed scheme attains the SPA-resistant scheme with any size of the pre-computed table. The designer of ECC can flexibly choose the table size suitable for the smartcards.

This paper is organized as follows: In Section 2 we review the scalar multiplication of elliptic curves. The width-w NAF and the fractional window method

are reviewed. In Section 3 the side channel attacks are discussed. The fast and memory efficient countermeasures are presented. In Section 4 we show the proposed scheme. The security and efficiency are discussed. In Section 5 we conclude the result of our paper.

2 Scalar Multiplication of ECC

In this section we review the scalar multiplication of elliptic curve cryptosystem (ECC). The width-w non-adjacent form (NAF) and the fractional window method are discussed

The scalar multiplication computes dP for a point P on the elliptic curve and a scalar d. A lot of algorithms of computing the scalar multiplication have been proposed. Because the inverse of a point P can be computed with little additional cost, the signed representation of d is usually deployed. The fastest method with less memory is the width-w non-adjacent form (NAF). The width-w NAF represents an n-bit integer $d = \sum_{i=0}^{n} d_w[i]2^i$, where $d_w[i]$ are odd integers with $|d_w[i]| < 2^{w-1}$ and there are at most one non-negative digit among w-consecutive digits. Therefore, we pre-compute the table with points $P, 3P, .., (2^{w-1}-1)P$, which has 2^{w-2} points including base point P. The points with the opposite sign are generated on the fly during the scalar multiplication.

Generating_Width-w_NAF
INPUT An n-bit d, a width w
OUTPUT $d_w[n], d_w[n-1], ..., d_w[0]$
1. $i \leftarrow 0$
2. While $d > 0$ do the following
2.1. if d is odd then do following
2.1.1. $d_w[i] \leftarrow d$ mods 2^w
2.1.2. $d \leftarrow d - d_w[i]$
2.2. else $d_w[i] \leftarrow 0$
2.3. $d \leftarrow d/2$, $i \leftarrow i+1$
3: Return $d_w[n], d_w[n-1], ..., d_w[0]$

Scalar_Multiplication_with_Width-w_NAF
INPUT $d_w[i]$, P, $(\|d_w[i]\|)P$
OUTPUT dP
1. $Q \leftarrow d_w[c]P$ for the largest c with $d_w[c] \neq 0$
2. For $i = c-1$ to 0
2.1. $Q \leftarrow$ ECDBL(Q)
2.2. if $d_w[i] \neq 0$ then $Q \leftarrow$ ECADD$(Q, d_w[i]P)$
3. Return Q

Several methods for generating the width-w NAF have been proposed [KT92], [MOC97], [BSS99], [Sol00]. Generating_Width-w_NAF is an algorithm that generates the width-w NAF proposed by Solinas [Sol00]. Notation "mods 2^w" at Step 2.1.1 stands for the signed residue modulo 2^w, namely $\pm 1, \pm 3, .., \pm(2^{w-1}-1)$. Note that the next $(w-1)$ consecutive bits of non-zero bits in the width-w NAF are always zero. It is known that the density of the non-zero bits of the width-w NAF is asymptotically equal to $1/(1+w)$.

Scalar_Multiplication_with_Width-w_NAF is an algorithm of computing the scalar multiplication using the width-w NAF. It is calculated from the most significant bit — elliptic curve doubling (ECDBL) at Step 2.1 is executed for each bit and elliptic curve addition (ECADD) at Step 2.2 is executed if and only if $d_w[i]$ is non-zero. Therefore we have to compute $(c+1)$-time ECDBLs and

$(c+1)/(1+w)$-time ECADDs, where c is the largest integer with $d_w[c] \neq 0$. If we choose larger width w, then the scalar multiplication becomes faster, but with more memory.

2.1 Fractional Width-w NAF

The width-w NAF uses the table $P, 3P, .., (2^{w-1}-1)P$. The size of the table takes discrete values $1, 2, 4, 8, ...$ for $w = 2, 3, 4, ...$ The density of non-zero bits of the width-w NAF also takes the discrete values $1/(w+1)$. In order to interpolate their intermediate values, Möller discussed how to construct the NAF with fractional widths [Möl02b]. His idea is to utilize the degenerated width-w NAF — some table values of the width-w NAF are not pre-computed[1], where $w > 3$. We call it the fractional width-w NAF in this paper.

The fractional width-w NAF can be easily generated by modifying Generating_Width-w_NAF. Indeed we insert the following step between Step 2.1.1 and Step 2.1.2:

if $|d_w[i]| > 2^{w-2} + B$ then $d_w[i] \leftarrow d_w[i] \bmod 2^{w-1}$,

where B is an integer $0 \leq B \leq 2^{w-2}$ that determines the table size and the efficiency between width-w and width-$(w-1)$ NAF. If we choose $B = 0$ or $B = 2^{w-2}$, then it becomes the width-$(w-1)$ or width-w NAF, respectively.

We define the width-w suitable for our paper in the following. Let $w = (w_0 - 1) + w_1$, where $w_0 - 1$ and w_1 are the integral and fractional parts[2] of w, respectively; $w_0 = \lceil w \rceil, w_1 = w - (w_0 - 1)$. The fractional part w_1 takes one of $1/2^{w_0-2}, 2/2^{w_0-2}, ..., (2^{w_0-2}-1)/2^{w_0-2}, 2^{w_0-2}/2^{w_0-2}$. Here the pre-computed points are $P, 3P, .., (2^{w_0-1}-1)P, (2^{w_0-1}+1)P, ..., (2^{w_0-1}+w_1 2^{w_0-1}-1)P$. There are $(1+w_1)2^{w_0-2}$ points. The non-zero density of the fractional width-w NAF is $1/(1+w)$. The scalar multiplication using the fractional width-w NAF is computed as same for Scalar_Multiplication_with_Width-w_NAF.

3 Side Channel Attacks and Their Countermeasures

In this section we review side channel attacks and their countermeasures.

Side channel attacks (SCA) are allowed to access the additional information linked to the operations using the secret key, e.g., timings, power consumptions,

[1] Strictly speaking, Möller's idea is as follows: Some values of the width-$(w+1)$ NAF are appended to the table. This enhances the speed but additional memory is required. On the contrary, the degenerated width-w NAF provides efficient memory but reduces the speed. In other words, speed and memory have a trade-off relation. The expression in this paper is different from that in [Möl02b], however, they are equivalent in this point. We use the former for the sake of the description of the proposed scheme in the following sections.

[2] We may define $w = w_0 + w_1$, $w_0 = \lfloor w \rfloor, w_1 = w - w_0$. For the sake of simplicity and the easiness of the comparison between original and proposed schemes in the following sections, we use the notations of the former.

etc. The attack aims at guessing the secret key (or some related information). Scalar_Multiplication_with_Width-w_NAF can be broken by the SCA. It calculates the ECADD if and only if the i-th bit is not zero. The standard implementation of ECADD is different from that of ECDBL, and thus the ECADD in the scalar multiplication can be detected using SCA.

If the attacker is allowed to observe the side channel information only a few times, it is called the simple power analysis (SPA). If the attacker can analyze several side channel information using a statistical tool, it is called the differential power analysis (DPA). The standard DPA utilizes the correlation function that can distinguish whether a specific bit is related to the observed calculation. In order to resist DPA, we need to randomize the parameters of elliptic curves.

There are three standard randomizations [Cor99,JT01]: (1)the base point is masked by a random point, (2)the secret scalar is randomized with multiplier of the order of the curve. (3)the base point is randomized in the projective coordinate (or Jacobian coordinate). Some attacks or weak classes against each countermeasure have been proposed [Gou03,OS00]. However, if these randomization methods are simultaneously used, no attack is known to break the combined scheme. In other words, SPA-resistant schemes can be easily converted to be DPA-resistant ones using these randomizations.

On the contrary, there are some schemes which try to achieve the SPA- and DPA-resistance simultaneously without using the combinations, e.g. randomized window methods [Wal02a,IYTT02,LS01,OA01], etc. The security of these schemes causes many controversies — some of them have been broken [OS02a, Wal02b,Wal03a] or less secure than expected [Wal03b]. Therefore we are interested in the SPA-resistant schemes.

3.1 SPA-Resistant Methods

We review the SPA-resistant schemes of computing the scalar multiplication.

There are three different approaches to resist the SPA. We explain these schemes in the following. (1)We construct the indistinguishable addition formula [LS01,JQ01,BJ02]. (2)We use the addition formula that always computes ECADD and ECDBL for each bit [Cor99,OS00,FGKS02,IT02,BJ02]. (3)We generate the addition chain with fixed pattern [Möl01a,Möl01b,Möl02a,OT03].

(1)Whereas the indistinguishable addition formula conceals addition and doubling, the attacker can detect the number of additions and doublings in the computation. In other words, the indistinguishable addition formula pulls the SPA back to the timing attack. Hence, this type is imperfect. (2)Since the second type does not compute the pre-computed points, the memory consumption is small. In addition, if we are allowed to use a special form such as Montgomery-form, this type is the fastest. However, some international standards [ANSI,IEEE,NIST,SEC] do not support such a form. Without using the special form, this type is not so fast because it requires many ECADD operations. (3)The third type utilizes pre-computed points for speeding up the computation, since the pre-computed points reduce the number of ECADD operations. Whereas the large number of the pre-computed points achieves a

fast computation, it requires large memory for storing the points. Okeya and Takagi proposed an SPA-resistant addition chain with small memory, which is based on the width-w NAF [OT03]. The algorithm is as follows (We modify it suitable for our proposed scheme):

SPA-resistant_Width-w_NAF_with_Odd_Scalar

INPUT An odd n-bit d
OUTPUT $d_w[n], d_w[n-1], ..., d_w[0]$

1. $r \leftarrow 0$, $i \leftarrow 0$, $r_0 \leftarrow w$
2. While $d > 1$ do the following
   2.1. $u[i] \leftarrow (d \bmod 2^{w+1}) - 2^w$
   2.2. $d \leftarrow (d - u[i])/2^{r_i}$
   2.3. $d_w[r + r_i - 1] \leftarrow 0, d_w[r + r_i - 2] \leftarrow 0, ..., d_w[r + 1] \leftarrow 0, d_w[r] \leftarrow u[i]$
   2.4. $r \leftarrow r + r_i$, $i \leftarrow i + 1$, $r_i \leftarrow w$
3. $d_w[n] \leftarrow 0, ..., d_w[r + 1] \leftarrow 0, d_w[r] \leftarrow 1$
4. Return $d_w[n], d_w[n-1], ..., d_w[0]$

The algorithm generates the SPA-resistant chain only for odd scalar, and the treatment for even scalar was discussed in [OT03]. We assume that the scalar d is odd in the following. At Step 2.1, the integer $u[i]$ is assigned as $(d \bmod 2^{w+1}) - 2^w$. The computation assures that $u[i]$ is odd whenever d is odd. Since $d - u[i] = d - (d \bmod 2^{w+1}) + 2^w = 2^w \bmod 2^{w+1}$, the resultant $(d - u[i])/2^w$ is odd. Thus, each integer $u[i]$ is odd. Note that d terminates with $d = 1$. Hence we can achieve the SPA-resistant chain, e.g., the fixed pattern

$$|\underbrace{0..0}_{w-1}x|\underbrace{0..0}_{w-1}x|...|\underbrace{0..0}_{w-1}x| \text{ with odd integers } |x| < 2^w.$$

The number of the pre-computed points is 2^{w-1}, and the density of the non-zero bit is $1/w$. The scalar multiplication using this chain is computed as same for Scalar_Multiplication_with_Width-w_NAF.

Note that this scheme is optimal in respect of the memory, and the table size takes $1, 2, 4, 8, ...$ for $w = 1, 2, 3, 4, ...$ If the designer of smart cards allows to use the table sizes $1, 2, 4, 8, ...$, this scheme is one of the best solutions. However, if he allows to use just the sizes $3, 5, 6, ...$ not $1, 2, 4, 8, ...$, it compromises the memory and/or the speed. This situation often occurs because some restrictions about resources such as memory and cost are determined by the applications of the smart cards, not the specifications of the cryptographic schemes. Such restrictions impose the flexibility on the cryptographic schemes. Hence, we need to construct such a scheme.

4 Proposed Scheme

In this section we propose a new SPA-resistant scheme with *any* table size. After describing its main idea, we present the details of our algorithm. We then discuss the security, the efficiency, and the memory requirement of our proposed scheme.

4.1 Main Idea

We describe the main idea of our proposed algorithm. The proposed scheme is converted from SPA-resistant_Width-w_NAF_with_Odd_Scalar using the idea of the fractional window method.

First, we discuss the security of the straight-forwardly combined scheme between Okeya-Takagi scheme [OT03] and the fractional window method [Möl02b], and find that this combined scheme is *not* secure against SPA. For the sake of simplicity we explain it with $w = 4$. SPA-resistant_Width-w_NAF_with_Odd_Scalar with $w = 4$ pre-computes the signed odd integer modulo 2^w, i.e., $U_w = \{\pm 1, \pm 3, \pm 5, \pm 7, \pm 9, \pm 11, \pm 13, \pm 15\}$. The fractional width-$w$ NAF can reduce it to smaller one, for instance, $F = \{\pm 1, \pm 3, \pm 5, \pm 7, \pm 9\}$. Note that it still contains the representations of the smaller modulus 2^{w-1}, i.e., $U_{w-1} = \{\pm 1, \pm 3, \pm 5, \pm 7\}$. The fractional window using class F is constructed by inserting the following step between Step 2.1 and 2.2:

$$\text{if } |u[i]| > 2^{w-1} + B, \quad \text{then } u[i] \leftarrow (u[i] \bmod 2^w) - 2^{w-1},\ r_i \leftarrow w - 1,$$

where B is an integer $0 < B < 2^{w-1}$ (in the case of F we choose $B = 1$). However, sequence $d_w[n], d_w[n-1], ..., d_w[0]$ generated by this fractional window method has no fixed pattern, so that it is not secure against the SPA. Indeed, we know $|u[i]| > 2^{w-1} + B$ if and only if $(w-2)$-consecutive zeros (i.e. $r_i = w-1$) appear. In order to overcome this bias, we propose two novel ideas in the following.

The first one is to control the choice of two moduli 2^w and 2^{w-1} as the probabilistic process. We reduce $u[i]$ with the uniform probability from the view point of the SPA. Since $u[i]$ with $|u[i]| < 2^{w-1}$ is possible to utilize both moduli 2^w and 2^{w-1}, the use of the following trick achieves our aim:

$$\begin{aligned}&\text{If } |u[i]| < 2^{w-1},\\ &\text{then } u[i] \leftarrow (u[i] \bmod 2^w) - 2^{w-1},\ r_i \leftarrow w - 1 \text{ with probability } 1 - P_w,\end{aligned}$$

where P_w is the probability that $u[i]$ within U_{w-1} remains the representation of mod 2^w, and we should select $P_w = \frac{\#F - \#U_{w-1}}{\#U_w - \#U_{w-1}}$. This means that we reduce $u[i]$ to the representation of mod 2^{w-1} with the same probability for both $|u[i]| < 2^{w-1}$ and $|u[i]| > 2^{w-1}$. Thus the SPA cannot distinguish the two distributions.

The second idea is to use a different representation of residue class modulo 2^w. The use of the different representation conceals the information that a specific $u[i]$ belongs to the class F. Instead of the "integer" B, we use the "subset" B of $U_w \setminus U_{w-1}$. Then the class F is chosen as $F = U_{w-1} \cup B$, since F contains any odd signed residue modulo 2^{w-1}. Because of $\#F = 10$ we should choose $\#B = 2$. Thus B is randomly chosen from one of $\pm 9, \pm 11, \pm 13$, or ± 15. The attacker cannot guess the value of B because of this random choice.

4.2 Proposed Algorithm

We present the algorithm of our proposed algorithm. The algorithm generates an SPA-resistant fractional width-w NAF for given n-bit odd scalar d and width w. The algorithm is as follows:

SPA-resistant_Fractional_Width-w_NAF_with_Odd_Scalar

INPUT An odd n-bit d, and a width w

OUTPUT $d_w[n+w_0-1], ..., d_w[n-1], ..., d_w[0]$, and $B = \{\pm b_1, ..., \pm b_{w_1 2^{w_0-2}}\}$

1. $w_0 \leftarrow \lceil w \rceil$, $w_1 \leftarrow w - (w_0 - 1)$
2. Randomly choose distinct integers
   $b_1, ..., b_{w_1 2^{w_0-2}} \in_R U^+_{w_0} \setminus U^+_{w_0-1} = \{2^{w_0-1}+1, 2^{w_0-1}+3, ..., 2^{w_0}-1\}$,
   and put $B = \{\pm b_1, ..., \pm b_{w_1 2^{w_0-2}}\}$, $P_w \leftarrow w_1$,
   where $U^+_v = \{1, 3, 5, ..., 2^v - 1\}$ for positive integer v.
3. $r \leftarrow 0$, $i \leftarrow 0$
4. While $d > 1$ do the following
   4.1. $x[i] \leftarrow (d \bmod 2^{w_0+1}) - 2^{w_0}$, $y[i] \leftarrow (d \bmod 2^{w_0}) - 2^{w_0-1}$
   4.2. if $|x[i]| < 2^{w_0-1}$ then
   $r_i \leftarrow w_0$, $u[i] \leftarrow x[i]$ with P_w; $r_i \leftarrow w_0 - 1$, $u[i] \leftarrow y[i]$ with $1 - P_w$
   else if $x[i] \in B$ then
   $r_i \leftarrow w_0$, $u[i] \leftarrow x[i]$ else $r_i \leftarrow w_0 - 1$, $u[i] \leftarrow y[i]$
   4.3. $d \leftarrow (d - u[i])/2^{r_i}$
   4.4. $d_w[r + r_i - 1] \leftarrow 0, d_w[r + r_i - 2] \leftarrow 0, ..., d_w[r + 1] \leftarrow 0, d_w[r] \leftarrow u[i]$
   4.5. $r \leftarrow r + r_i$, $i \leftarrow i + 1$
5. $d_w[n + w_0 - 1] \leftarrow 0, ..., d_w[r + 1] \leftarrow 0, d_w[r] \leftarrow 1$
6. Return $d_w[n+w_0-1], ..., d_w[n-1], ..., d_w[0]$, and $B = \{\pm b_1, ..., \pm b_{w_1 2^{w_0-2}}\}$

At Step 1 we assign the integral part w_0 and the fractional part w_1 of the width w. At Step 2 the pre-computed index $b_1, ..., b_{w_1 2^{w_0-2}}$ that belongs to upper set $U^+_{w_0} \setminus U^+_{w_0-1}$ are randomly chosen. The random signed index B is returned as the part of output. The reduction probability P_w is assigned. At Step 3 integers r, i are initialized. Step 4 is the main loop of the proposed algorithm. At Step 4.1 we generate two different residue values $x[i] \bmod 2^{w_0}$ and $y[i] \bmod 2^{w_0-1}$. At Step 4.2 one of $x[i]$ and $y[i]$ is assigned for $u[i]$ based on both the size $|x[i]|$ and the probability P_w. At Step 4.3 we eliminate least r_i bits of d. At Step 4.4 bit information $d_w[i]$ is assigned. The $(r_i - 1)$ consecutive bits after the lowest bit $d_w[r]$ are zero. At Step 4.5 integers r, i are updated. Finally we return all bits of the proposed addition chain. The total bit could be at most w_0 bits larger than the original n bits.

The pre-computed points are calculated using not only a base point P and a width w but also the randomized index $B = \{b_1, ..., b_{w_1 2^{w_0-2}}\}$. For index B the pre-computed points are $P, 3P, ..., (2^{w_0-1} - 1)P$ and $b_1 P, ..., b_{w_1 2^{w_0-2}} P$. The scalar multiplication using the proposed chain is computed as same for Scalar_Multiplication_with_Width-w_NAF.

At Step 4.2 $u[i] = x[i]$ is assigned with probability $P_w = a/2^{w_0-2}$ for $a = 1, 2, .., 2^{w_0-2}$. We can easily generate the "probability" using a 1-bit random number generator as follows: First we obtain a random $(w_0 - 2)$-bit number *rand* by executing the 1-bit random number generator $w_0 - 2$ times. Then we assign $u[i] = x[i]$ if and only if $rand \leq a$ holds. The probability of $rand \leq a$ is exactly $a/2^{w_0-2}$ due to the uniform distribution of *rand* in $\{0, 1, ..., 2^{w_0-1} - 1\}$. A 1-bit random number generator is usually equipped on smart cards. We can generate the probability P_w with a small additional cost.

4.3 Security against SPA

We discuss the security of the proposed scheme against the SPA. We prove that the sequence $d_w[n], d_w[n-1], ..., d_w[0]$ arisen from the proposed algorithm has no correlation to the secret bit information in the sense of SPA.

Theorem 1. *The proposed scheme is secure against the SPA.*

Proof. $u[i]$ is a non-zero odd integer from the construction of the proposed algorithm. Thus, any subsequence of the consecutive zero bits in the sequence $d_w[n], d_w[n-1], ..., d_w[0]$ has the length $w_0 - 1$ or $w_0 - 2$;

$$..0u[i+1]\Big|\underbrace{0..0}_{r_i-1}u[i]\Big|00.., \quad r_i = w_0 \text{ or } w_0 - 1.$$

The corresponding AD sequence is

$$..DDA\Big|\underbrace{D..DD}_{r_i}A\Big|DD.., \quad r_i = w_0 \text{ or } w_0 - 1,$$

where A and D indicate ECADD and ECDBL, respectively. Hence, all the information that the SPA can obtain from the AD sequence is the length of the consecutive zero, namely r_i.

In the following we prove that r_i provides no information about the secret scalar d. Indeed we show that two AD sequences $\underbrace{D..DD}_{w_0}A$ and $\underbrace{D..DD}_{w_0-1}A$ are independently distributed from the secret scalar d. Here we can assume that d is randomly and uniformly distributed in all n-bit odd integers, because d is the secret key. Since $x[i]$ and $y[i]$ are assigned dependently on only the lower (w_0+1) bits of d, they are random w_0-bit and (w_0-1)-bit signed odd integers, respectively, due to the uniform distribution of d. Thus, we consider the lower (w_0+1) bits of the binary representations of d. Note that the lowest bit is always 1, and is converted by the preceding d. Thus, we do not need to consider the effect of the lowest bit.

We estimate the probability that $x[i]$ or $y[i]$ is assigned at Step 4.2 of the proposed algorithm. We have the following 4 cases: $\mathrm{LSB}_{w_0+1}(d) = 00 * ... * 1, 01 * ... * 1, 10 * ... * 1$, and $11 * ... * 1$, where $\mathrm{LSB}_{w_0+1}(d)$ denotes the lower (w_0+1) bits of d. First we discuss the case that $\mathrm{LSB}_{w_0+1}(d) = 00 * ... * 1$. In this case we have $-2^{w_0} < x[i] < -2^{w_0-1}$. That is, $|x[i]| \geq 2^{w_0-1}$. At Step 4.2, the lower half instructions are executed. Since the probability of $x[i] \in B$ is $\#B/(\#U_{w_0} - \#U_{w_0-1}) = P_w$, we have $r_i = w_0$ with the probability P_w, and $r_i = w_0 - 1$ with the probability $1 - P_w$. Next, we discuss the case of $\mathrm{LSB}_{w_0+1}(d) = 01 * ... * 1$. At Step 4.2, the upper half instructions are executed, since $-2^{w_0-1} < x[i] < 0$. Thus we have $r_i = w_0$ with the probability P_w, and $r_i = w_0 - 1$ with the probability $1 - P_w$. In the case of $\mathrm{LSB}_{w_0+1}(d) = 10 * ... * 1$, the upper half instructions at Step 4.2 are executed, since $0 < x[i] < 2^{w_0-1}$. Thus we have $r_i = w_0$ with the probability P_w, and $r_i = w_0 - 1$ with the probability

$1-P_w$. Finally, in the case of $\mathrm{LSB}_{w_0+1}(d) = 11*...*1$, the lower half instructions at Step 4.2 are executed, since $2^{w_0-1} < x[i] < 2^{w_0}$. Because the probability that $x[i] \in B$ is P_w, we have $r_i = w_0$ with the probability P_w, and $r_i = w_0 - 1$ with the probability $1 - P_w$. Therefore, the proposed scheme produces $r_i = w_0$ with probability P_w and $r_i = w_0 - 1$ with probability $1 - P_w$, which is independent from d. □

We point out that each bit $d_w[i]$ is not randomly distributed in class $U_{w_0-1} \cup B$. The auxiliary variables $x[i]$ and $y[i]$ of the proposed algorithm are randomly distributed in U_{w_0} and U_{w_0-1}, respectively. The resulting $d_w[i]$ is assigned as $x[i]$ with probability P_w and $y[i]$ with probability $1 - P_w$, respectively. Thus some points in class $U_{w_0-1} \cup B$ appear with higher probability. However, the proposed scheme conceals such points, because B is randomly chosen and points in U_{w_0-1} are also randomly chosen. That is, the attacker might reveal the distribution, however he/she cannot detect the correspondence between the point with higher probability and the value of $d_w[i]$.

On the contrary, if we simply choose predetermined numbers like the fractional window method, then the scheme is not secure against SPA. For example, we consider the case of $w = 3 + 1/8$; $B = \{\pm 9\}$. If the length of the consecutive zero bit is 3, then the conditional probabilities that the next non-zero $d_w[i] = \pm 9$ are $1/4$ each, while the probabilities that $d_w[i] = \pm 1, \pm 3, \pm 5, \pm 7$ are $1/16$ each. Thus, the probabilities are not uniform. Since the attacker knows the predetermined numbers that belong to B, he/she has an advantage to guess $d_w[i]$. For example, the use of the attack proposed by Oswald [Osw02] reduces the cost of the exhaustive search for the candidates of the secret key which are not uniformly distributed.

4.4 Memory Consumption and Computation Cost

We discuss the memory and efficiency of the proposed scheme.

The efficiency of ECC is strongly depending on the representation of the base fields, the coordinate systems, and the definition equations. The proposed scheme aims at developing a secure encoding of the addition chain, and it can freely choose these parameters. We attach importance to the flexibility of cryptographic schemes, so that we estimate no computation cost of individual optimizations. We intend to estimate the trade-off between memory consumption (the size of pre-computed table) and the computation cost (the density of non-zero bits) for the proposed scheme. We have the following theorem.

Theorem 2. *The size of the pre-computed table is* $(1 + w_1)2^{w_0-2}$. *The density of the non-zero bits is asymptotically* $\frac{w_0-P_w}{(w_0-1)w_0}$.

Proof. In Step 2 we pre-compute set B whose size is exactly equal to $w_1 2^{w_0-2}$. In addition to the set B, the proposed scheme prepares the pre-computed points $P, 3P, ..., (2^{w_0-1}-1)P$ for the base point P, the number of points is 2^{w_0-2}. Thus the number of all the pre-computed points is $(1 + w_1)2^{w_0-2}$.

In Step 4.2 we assign the length of the consecutive zero bit is always $w_0 - 1$ (i.e. $\underbrace{0..0}_{w_0-1} u[i]$) with probability P_w or $w_0 - 2$ (i.e. $\underbrace{0..0}_{w_0-2} u[i]$) with probability $1 - P_w$. Therefore the density of non-zero bits is asymptotically $\frac{w_0 - P_w}{(w_0-1)w_0}$. □

In Table 1 we summarize these results. The table size includes the base point P itself. If w is integral; $w = w_0$, then the size of the table and the density of non-zero bit of the proposed scheme are same as those of the scheme proposed by Okeya-Takagi, respectively. The proposed scheme interpolates the gap of the discrete table sizes 2^{w-1} for $w = 2, 3, 4, ...$, namely all possible numbers of pre-computed table can be used. Thus the designers of elliptic curve cryptosystems can flexibly choose the table size suitable for the smart card.

Table 1. Memory and Efficiency of the Proposed Scheme

Width	2	**2.5**	3	**3.25**	**3.5**	**3.75**	4	**4.125**	···
Table Size	2	**3**	4	**5**	**6**	**7**	8	**9**	···
Non-Zero Density	0.5	**0.42**	0.33	**0.313**	**0.291**	**0.271**	0.25	**0.244**	···

4.5 Other Security Properties

We discuss other security properties of our proposed scheme, namely a possible attack using the DPA and a security comparison with the randomized window methods.

The proposed scheme aims at resisting the SPA, but we discuss the security against the DPA. In Section 3, we mentioned that the SPA-resistant schemes can be easily converted to be DPA-resistant ones using randomization tricks [Cor99, JT01]. Thus, the proposed scheme can be converted to be DPA-resistant one. On the other hand, the window methods using the fixed secret scalar are vulnerable to the sophisticated DPA, e.g., the second order DPA [OS02b,OT03] and the address-bit DPA [IIT02]. These DPA can detect which pre-computed points are called for the ECADD, and the associated bits of the secret scalar can be revealed. Since also countermeasures against such sophisticated DPA attacks were proposed in the papers [OS02b,OT03,IIT02], the combined scheme is secure against the sophisticated DPA.

The addition chain of the proposed scheme is generated by randomly choosing two window lengths 2^{w_0} and 2^{w_0-1}. There are several window methods that intend to protect the DPA by randomizing the addition chain [Wal02a,IYTT02, LS01,OA01], etc. The goal of these schemes is different from ours, but we compare the security in the sense of the SPA. These schemes produce several AD sequences depending on the secret scalar d and random numbers. However, the distribution of the AD sequences are not uniform, but depends on the *secret* scalar. Indeed, some of them were broken [OS02a,Wal02b,Wal03a,OS03,HCJ+03]

because of such bias. On the other hand, the randomization of the proposed scheme is independent from the *secret* scalar. There are two different AD sequences for the proposed scheme, and the probability of appearing the two AD sequences only depends on the width-w, which is a *public* parameter.

5 Conclusion

We proposed an SPA-resistant scalar multiplication for elliptic curve cryptosystem, which allows us to choose any size of the pre-computation points with efficient running time.

It is expected that smartcards are able to equip highly functional applications. In addition, in order to accomplish the aims of users, some applications are often appended to (deleted from) the smartcards. The memory space of cryptographic functions depends on these applications. In other words, the cryptographic schemes are imposed on the flexibility of the memory consumption and efficiency. Indeed, with our proposed scheme, (1)the designer of smart cards can flexibly choose the table size suitable for the individual situations, (2)the private information in the smart cards are protected against the side channel attacks.

References

[ANSI] ANSI X9.62, Public Key Cryptography for the Financial Services Industry, *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, (1998).

[BJ02] Brier, É., Joye, M., *Weierstrass Elliptic Curves and Side-Channel Attacks*, Public Key Cryptography (PKC2002), LNCS 2274, (2002), 335–345.

[BSS99] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, Cambridge University Press, 1999.

[Cor99] Coron, J.S., *Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems*, Cryptographic Hardware and Embedded Systems (CHES'99), LNCS 1717, (1999), 292–302.

[FGKS02] Fischer, W., Giraud, C., Knudsen, E.W., Seifert, J.P., *Parallel scalar multiplication on general elliptic curves over* $\mathbf{F}_p$ *hedged against Non-Differential Side-Channel Attacks*, International Association for Cryptologic Research (IACR), Cryptology ePrint Archive 2002/007, (2002). `http://eprint.iacr.org/2002/007/`

[Gou03] Goubin, L., *A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems*, Public Key Cryptography, (PKC 2003), LNCS 2567, (2003), 199–211.

[HaM02] Ha, J., and Moon, S., *Randomized Signed-Scalar Multiplication of ECC to Resist Power Attacks*, Workshop on Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), LNCS 2523, (2002), 551–563.

[HCJ+03] Dong-Guk Han, Nam Su Chang, Seok Won Jung, Young-Ho Park, Chang Han Kim, Heuisu Ryu, *Cryptanalysis of the Full version Randomized Addition-Subtraction Chains*, to appear in ACISP 2003.

[IIT02] Itoh, K., Izu, T., and Takenaka, M., *Address-bit Differential Power Analysis on Cryptographic Schemes OK-ECDH and OK-ECDSA*, Workshop on Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), LNCS 2523, (2002), 129–143.

[IYTT02] Itoh, K., Yajima, J., Takenaka, M., and Torii, N., *DPA Countermeasures by improving the Window Method*, Workshop on Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), LNCS 2523, (2002), 318–332.

[IEEE] IEEE P1363, Standard Specifications for Public-Key Cryptography. `http://groupe.ieee.org/groups/1363/`

[IT02] Izu, T., Takagi, T., *A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks*, Public Key Cryptography (PKC2002), LNCS 2274, (2002), 280–296.

[JQ01] Joye, M., Quisquater, J.J., *Hessian elliptic curves and side-channel attacks*, Cryptographic Hardware and Embedded Systems (CHES'01), LNCS 2162, (2001), 402–410.

[JT01] Joye, M., Tymen, C., *Protections against differential analysis for elliptic curve cryptography: An algebraic approach*, Cryptographic Hardware and Embedded Systems (CHES'01), LNCS2162, (2001), 377–390.

[Kob87] Koblitz, N., *Elliptic curve cryptosystems*, Math. Comp. 48, (1987), 203–209.

[KJJ99] Kocher, C., Jaffe, J., Jun, B., *Differential Power Analysis*, Advances in Cryptology – CRYPTO '99, LNCS 1666, (1999), 388–397.

[KT92] K. Koyama and Y. Tsuruoka, *Speeding Up Elliptic Curve Cryptosystems using a Signed Binary Windows Method*, Advances in Cryptology – CRYPTO '92, LNCS 740, (1992), 345–357.

[LS01] Liardet, P.Y., Smart, N.P., *Preventing SPA/DPA in ECC systems using the Jacobi form*, Cryptographic Hardware and Embedded System (CHES'01), LNCS2162, (2001), 391–401.

[Mil86] Miller, V.S., *Use of elliptic curves in cryptography*, Advances in Cryptology - CRYPTO '85, LNCS218, (1986), 417–426.

[MOC97] Atsuko Miyaji, Takatoshi Ono, Henri Cohen, *Efficient elliptic curve exponentiation*, Information and Communication Security (ICICS 1997), (1997), 282–291.

[Möl01a] Möller, B., *Securing Elliptic Curve Point Multiplication against Side-Channel Attacks*, Information Security (ISC2001), LNCS2200, (2001), 324–334.

[Möl01b] Möller, B., *Securing elliptic curve point multiplication against side-channel attacks*, addendum: Efficiency improvement. `http://www.informatik.tu-darmstadt.de/TI/Mitarbeiter/moeller/ecc-scaisc01.pdf`, (2001).

[Möl02a] Möller, B., *Parallelizable Elliptic Curve Point Multiplication Method with Resistance against Side-Channel Attacks*, Information Security Conference (ISC 2002), LNCS 2433, (2002), 402–413.

[Möl02b] Möller, B., *Improved Techniques for Fast Exponentiation*, The 5th International Conference on Information Security and Cryptology (ICISC 2002), LNCS 2587, (2003), 298–312.

[NIST] National Institute of Standards and Technology, FIPS 186-2, `http://csrc.nist.gov/publication/fips/fips186-2/fips186-2.pdf`

[OA01] Oswald, E., Aigner, M., *Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks*, Cryptographic Hardware and Embedded Systems (CHES'01), LNCS2162, (2001), 39–50.

[OS00] Okeya, K., Sakurai, K., *Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack*, Progress in Cryptology - INDOCRYPT 2000, LNCS1977, (2000), 178–190.

[OS02a] Okeya, K., Sakurai, K., *On Insecurity of the Side Channel Attack Countermeasure using Addition-Subtraction Chains under Distinguishability between Addition and Doubling*, The 7th Australasian Conference in Information Security and Privacy, (ACISP 2002), LNCS2384, (2002), 420–435.

[OS02b] Okeya, K., Sakurai, K., *A Second-Order DPA Attack Breaks a Window-method based Countermeasure against Side Channel Attacks*, Information Security Conference (ISC 2002), LNCS 2433, (2002), 389–401.

[OS03] Okeya, K., Sakurai, K., *A Multiple Power Analysis Breaks the Advanced Version of the Randomized Addition-Subtraction Chains Countermeasure against Side Channel Attacks*, in 2003 IEEE Information Theory Workshop (ITW 2003) (these proceedings), (2003).

[Osw02] Oswald, E., *Enhancing Simple Power-Analysis Attacks on Elliptic Curve Cryptosystems*, Workshop on Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), LNCS 2523, (2002), 82–97.

[OT03] Okeya, K., Takagi, T., *The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks*, Topics in Cryptology, The Cryptographers' Track at the RSA Conference 2003 (CT-RSA 2003), LNCS2612, (2003), 328–342.

[SEC] Standards for Efficient Cryptography Group (SECG), `http://www.secg.org`

[Sol00] Solinas, J.A., *Efficient Arithmetic on Koblitz Curves*, Design, Codes and Cryptography, 19, (2000), 195–249.

[Wal02a] Walter, C.D., *Some Security Aspects of the Mist Randomized Exponentiation Algorithm*, Workshop on Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), LNCS 2523, (2002), 564–578.

[Wal02b] Walter, C.D., *Breaking the Liardet-Smart Randomized Exponentiation Algorithm*, Proceedings of CARDIS'02, USENIX Assoc, (2002), 59–68.

[Wal03a] Walter, C.D., *Security Constraints on the Oswald-Aigner Exponentiation Algorithm*, International Association for Cryptologic Research (IACR), Cryptology ePrint Archive 2003/013, (2003). `http://eprint.iacr.org/2003/013/`

[Wal03b] Walter, C.D., *Seeing through Mist Given a Small Fraction of an RSA Private Key, Topics in Cryptology*, Topics in Cryptology, The Cryptographers' Track at the RSA Conference 2003 (CT-RSA 2003), LNCS2612, (2003), 391–402.