Enhanced Doubling Attacks on Signed-All-Bits Set Recoding

HeeSeok Kim[1], Tae Hyun Kim[1], Jeong Choon Ryoo[1],
Dong-Guk Han[2⋆], Ho Won Kim[2], and Jongin Lim[1]

[1] Graduate School of Information Management and Security, Korea University, Korea
{heeseokkim, thkim, jcryoo, jilim}@cist.korea.ac.kr
[2] Electronics and Telecommunications Research Institute(ETRI), Korea
{christa, khw}@etri.re.kr

Abstract. In cryptographic devices like a smart card whose computing ability and memory are limited, cryptographic algorithms should be performed efficiently. However, the issue of efficiency sometimes raises vulnerabilities against side channel attacks (SCAs). In elliptic curve cryptosystems, one of main operations is the scalar multiplication. Thus it must be constructed in safety against SCAs. Recently, Hedabou et al. proposed a signed-all-bits set (sABS) recoding as simple power analysis countermeasure, which is also secure against doubling attack (DA). In this paper we propose enhanced doubling attacks which break Hedabou's countermeasure based on sABS recoding, and then show the statistical approach of noise reduction to experiment on the proposed attacks in actuality. We also introduce a countermeasure based on a projective coordinate.

Keywords : Side Channel Attacks, sABS recoding, SPA-based analysis, scalar multiplication, Doubling Attack

1 Introduction

Many designers of cryptosystems have proposed cryptographic algorithms based on theoretical security such as integer factoring problem and discrete logarithm problem. Even though these algorithms are proved as safe with mathematical tools, they could be vulnerable to physical attacks using additional information via side channel. Such type of attacks is referred to as Side Channel Attacks (SCAs) first introduced by Kocher [10]. In categories of SCAs, actively researched power analysis attack classifies into the simple power analysis (SPA) and the differential power analysis (DPA). To resist SPA among the power analysis attack, many researchers have proposed various countermeasures. Above all, two famous countermeasures are Coron's method [4] adding dummy operations and the scalar multiplication algorithm using singed all-bits set (sABS) recoding [7]. But, Coron's dummy method exposes a weakness by doubling attack (DA)

[⋆] This author is the corresponding author of this paper.

[5] introduced by Fouque et al. in 2003. Contrary to Coron's dummy method, sABS recoding based countermeasure is secure against doubling attack.

In this paper we propose two enhanced doubling attacks applicable to scalar multiplication algorithm based on sABS recoding, and introduce an experimental method to justify the practicality of the proposed attacks including experiment results. The one proposed attack is called recursive attack which is an analysis method finding the secret key from the most significant bit to the least significant bit in sequence through an adjustment of two input data to do the same elliptic curve doubling (ECDBL) in the vicinity of guessing secret bit. The other is called initializing attack which is an analysis method adjusting one input data to do fixed ECDBL at the guessing secret bit. We also propose a solution to detect equality of the compared two ECDBLs power signals by using a statistical approach of noise reduction when the noise is more important. Furthermore, we find that if we use a projective coordinate system to represent an elliptic curve element then it is secure against not only general doubling attack and but the proposed enhanced doubling attacks.

The remainder of this article is organized as follows. Section 2 represents that Coron's dummy method as an SPA countermeasure is vulnerable to SPA-based DA. Our proposed attacks are introduced in Section 3. These new attacks are SPA-based analysis methods applicable to scalar multiplication algorithm using sABS recoding among SPA countermeasures, and a practical attack method and a realistic possibility is showed in Section 4. Section 5 represents countermeasures on our attacks. Finally we conclude in Section 6.

2 Side Channel Attacks and Countermeasures

Since side channel attacks using additional information via side channel were introduced by Kocher, a various attack methods of this class have been proposed. There are fault insertion attack [2, 20], timing attack [10], power analysis attack [11, 12], electromagnetic emission attack [18], and so on. In power analysis attack, there are SPA that can expose secret key to be used by means of simple observation of a power consumption trace and DPA that analyzes multiple signals statistically without physical transformation of a smart card. DPA requires measuring a lot of power consumption and additional information such as description of implementation. But SPA is so simple. In this section, we introduce SPA and SPA countermeasures. Also we represent that Coron's dummy method known to be immune to SPA exposes a weakness against SPA-based DA.

2.1 Simple Power Analysis to ECC

Koblitz and Miller proposed elliptic curve cryptosystems (ECCs) in 1985 [9, 14]. Because of short length of the secret key for guaranteeing the same security with RSA, ECCs are suitable for mobile devices such as mobile phones, smart cards, and PDAs which are limited at storage space and bandwidth. While cryptosystems such as RSA [16] use the operation of modular exponentiation, ECCs use the operation of scalar multiplication that is regarded as similar method. And so this operation is the most dominant operation in ECCs. Scalar multiplication is to compute dP from a point P on an elliptic curve. Algorithm 1 that is a standard method for computing the scalar multiplication works by scanning the secret key from MSB to LSB.

Algorithm 1 Double-and-add algorithm

```
Input : A point $P$, and $d = (d_{n-1}d_{n-2}...d_1d_0)_2$ , $d_{n-1} = 1$
Output : $dP$
    1. $S = P$
    2. For $i = n - 2$ downto 0
        2.1 $S = 2S$
        2.2 If $d_i = 1$, $S = S + P$
    3. Return($S$)
```

In Algorithm 1, the scalar multiplication for secret key d is carried out by scanning from MSB to LSB. If a specific bit of d is 1, the algorithm comes into operation of Step 2.1, 2.2. If not, that comes into operation of Step 2.1 only. In other words, depending on the key bit value, one carries out both elliptic curve addition (ECADD) and ECDBL, the other carries out ECDBL only. In general, ECADD has different power consumption from ECDBL [3]. Thus we can deduce the secret key by a power consumption of the scalar multiplication. This method that can expose a portion of secret key using only one signal is called by SPA.

2.2 SPA countermeasures

Algorithms that have a conditional branch depending on the secret key are weak against SPA. For eliminating this weakness, algorithms that carry out unnecessary ECADDs regardless of the value of bit have been proposed.

SPA countermeasure 1 - Dummy Operation Algorithm 2 proposed as SPA countermeasure executes dummy operation when the value of bit is ‘0’.

Algorithm 2 Coron’s dummy method

```
Input : A point $P$, and $d = (d_{n-1}d_{n-2}...d_1d_0)_2$ , $d_{n-1} = 1$
Output : $dP$
    1. $S[0] = P$
    2. For $i = n - 2$ downto 0
        2.1 $S[0] = 2S[0]$
        2.2 $S[1] = S[0] + P$
        2.3 $S[0] = S[d_i]$
    3. Return($S[0]$)
```

This algorithm executes Step 2.2 at every loop regardless of the value of bit. In any loop that be carried out when the value of bit is 0, Step 2.2 is the superfluous ECADD operation. But in spite of this demerit in sense of efficiency, this algorithm is secure against SPA because it always compute both ECADD and ECDBL independent of a bit of the secret key. But this algorithm has problem for efficiency, moreover we will describe that that is insecure against SPA-based DA in the next subsection.

SPA countermeasure 2 - sABS recoding Other countermeasures are to change the binary representation of the secret key using signed digits. Among those countermeasures, sABS recoding method proposed by Hedabou et al. recodes the secret key into a new representation without zero bits by converting 00...01 into $1\bar{1}...\bar{1}\bar{1}$ where $\bar{1}$ means -1.

Algorithm 3 executes the scalar multiplication with this recoded representation. If the secret key is even, this algorithm carries out the operation of $dP = (d+1)P - P = t'P - P$ like Step 1 and Step 5 where t' is the sABS recoded value of $d+1$. Because this recoded value of the secret key does not have '0' bit, the sABS recoding method is secure against SPA: in Step 4 of Algorithm 3, ECADD or elliptic curve subtraction (ECSUB) whose power consumption is similar to ECADD is always carried out in every loop, and so this method does not come out the weakness against SPA. Note that we show that sABS recoding is secure against the original DA at the following section.

Algorithm 3 Scalar multiplication with sABS recoding

```
Input : A point $P$, and $d = (d_{n-1}d_{n-2}...d_1d_0)_2$ , $d_{n-1} = 1$
Output : $dP$
    1. if($d$ is even) then $t = d + 1$
    2. sABS recoded value of $t$ : $t' = (t'_{n-1}t'_{n-2}...t'_1t'_0)_2$, $t'_i \in \{-1, 1\}$
    3. $S = P$
    4. For $i = n - 2$ downto 0
        4.1 $S = 2S$
        4.2 if($t'_i = 1$) then $S = S + P$, else then $S = S - P$
    5. if($d$ is even) then $S = S - P$
    6. Return($S$)
```

2.3 Doubling Attack

Algorithm 2 exposes a weakness against DA that uses not DPA but SPA. Because DA is a SPA-based analysis, this attack method is much simpler than the existing DPA method.

Doubling Attack and Weakness of Coron's dummy method DA is a possible method when an attacker has an ability that if the card computes

ECDBL(A) and ECDBL(B), he is able to check whether $A = B$ or not, even so actual values of A and B don't be recovered by himself. The basic idea of this attack is like the table below:

Input	1	**0**	1	**0**	**0**	1	**0**	**0**	1
P	0	$2P$	**4P**	$10P$	**20P**	**40P**	$82P$	**164P**	**328P**
	P	$3P$	$5P$	$11P$	$21P$	$41P$	$83P$	$165P$	$329P$
$2P$	0	**4P**	8P	**20P**	**40P**	80P	**164P**	**328P**	656P
	$2P$	$6P$	$10P$	$22P$	$42P$	$82P$	$166P$	$330P$	$658P$

We compare power signals when the card computes dP and $d(2P)$ for input point P and $2P$, this recovers all bits of the secret key through confirmation of equal power consumption by same ECDBL in the vicinity of the bit value '0'.

Security of sABS recoding against Doubling Attack DA is the attack method using weakness that a certain bit of d is '0'. Hence, the original DA cannot be applied to the scalar multiplication with sABS recoding that does not have bit value '0'. In Algorithm 3, if an attacker tries DA to expect the value of t'_{n-l} that is the upper l-th bit of recoded value of the secret key (d or $d+1$), the values computed until the upper l-th bit for input point P and the upper $(l-1)$-th bit for input point $2P$ should be the same as the following equation.

$$(\sum_{i=0}^{l-1} t'_{n-l+i}2^i)P = (\sum_{i=0}^{l-2} t'_{n-l+i+1}2^i)2P \quad \Rightarrow \quad (\sum_{j=0}^{l-1} t'_{n-l+j}2^j)P = (\sum_{j=1}^{l-1} t'_{n-l+j}2^j)P. \quad (1)$$

Therefore, to satisfy equation (1), we know easily that $t'_{n-l} = 0$. However, because sABS recoded value is composed of '1' and '-1', this is not vulnerable to DA.

3 Proposed Attacks

Our paper proposes two attacks, recursive attack and initializing attack, against SPA countermeasure that executes the scalar multiplication using sABS recoding. Like original DA, these new two attacks are possible when an attacker has ability to decide whether $A = B$ or not when smart card computes ECDBL(A) and ECDBL(B). Our paper also offers an authenticity of this assumption through experimental result and theory in the next section. At first, we introduce our new attacks in this section.

3.1 Recursive Attack

In the proposed attack methods, recursive attack's basic idea is like follows: Suppose an attacker guesses a specific bit of the target secret key d, and he regulates two input values to have equal power consumption by the same ECDBL in the vicinity of the target bit. In this way, all bits of the secret key d can be discovered in sequence.

Input value regulation Suppose that an attacker knows upper bits $t'_{n-1}t'_{n-2}\cdots t'_{u+2}t'_{u+1}$ of $t' = (t'_{n-1}t'_{n-2}...t'_1t'_0)_2$, $t'_i \in \{-1, 1\}$, which is a recoded value of secret key d in Algorithm 3, let us regulate input values to find the value of t'_u. When we get two input values of xP and yP, we should regulate values of x and y for originating same ECDBL in phase of operation from $i = u$ to $i = u - 1$ for xP and from $i = u + 1$ to $i = u$ for yP in Step 4.1 of Algorithm 3. If we guess the value of t'_u as 1, the value of S until $i = u$ for input value xP is $S = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u} + 1)xP$, and the value of S until $i = u + 1$ for input value yP is $S = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u-1})yP$ in Step 4 of Algorithm 3. To originate the same ECDBL at this moment, if we select xP and yP satisfying this equation $(\sum_{i=u+1}^{n-1} t'_i 2^{i-u} + 1)xP = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u-1})yP$, we can get the following values.

$$xP = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u-1})P, \quad yP = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u} + 1)P \tag{2}$$

If we guess the value of t'_u as -1, xP and yP are the following values for the same reason as mentioned above.

$$xP = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u-1})P, \quad yP = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u} - 1)P \tag{3}$$

$\sum_{i=u+1}^{n-1} t'_i 2^{i-u-1}$ in equation (2) and (3) is a upper portion of t'_u that we are trying to find the bit of recoded value t' in Algorithm 3. Hence, if we name this value as k, two selected input values are kP and $(2k + 1)P$ (kP, $(2k - 1)P$) in the case that we guess the value of t'_u as 1 (-1). In this way, we can find all bits of the secret key d from MSB to LSB in sequence.

Scenario and Example of Recursive Attack In this section, we introduce the scenario of recursive attack for finding the entire value of the secret key, and then give a simple example to help understanding of this attack. Table 1 is the scenario of recursive attack for finding the entire information of the secret key.

Let us Consider this scenario. For example, if the secret key d is $(101010011)_2$ in Algorithm 3, then the value of t' becomes $11\bar{1}1\bar{1}1\bar{1}\bar{1}1$. In Table 1, suppose that the attacker already knows upper four bits of t' (upper 4 bits values : $11\bar{1}1 = (11)_{10}$). For now, he attempts to guess upper 5-th bit as 1. Hence, input values to know this bit are $11P$ used already to know the upper 4-th bit and $(2 * 11 + 1)P$ viewed in Step 4 of the scenario. And then he confirms as the table below whether the same ECDBL is originated in the vicinity of the upper 5-th bit or not.

Input	1	1	$\bar{1}$	1	$\mathbf{\bar{1}}$	1	$\bar{1}$	$\bar{1}$	1
$11P$	0 $11P$	$22P$ $33P$	$66P$ $55P$	$110P$ $121P$	$242P$ **231P**	**462P** $473P$	$\cdots$ $\cdots$	$\cdots$ $\cdots$	$\cdots$ $\cdots$
$23P$	0 $23P$	$46P$ $69P$	$138P$ $115P$	$230P$ **253P**	**506P** $483P$	$\cdots$ $\cdots$	$\cdots$ $\cdots$	$\cdots$ $\cdots$	$\cdots$ $\cdots$

```
Step 1. Set $k = 1, i = 2$.
Step 2. Measure a power consumption $C_1$ related with the input point $P$.
Step 3. If $i = n$, goto Step 7.
Step 4. Measure a power consumption $C_2$ related with the input point $(2k + 1)P$.
Step 5. If $C_1$ and $C_2$ have the same ECDBL signal in the vicinity of the upper
            $i$-th bit, then $k = 2k + 1$.
        Else then,
            a. Measure a power consumption $C_2$ related with input point $(2k - 1)P$.
            b. $k = 2k - 1$.
Step 6. $C_1 \longleftarrow C_2$ and $i = i + 1$ goto Step 3.
Step 7. If the output about the input point $P$ is $(2k + 1)P$, then return $2k + 1$.
        Else then, return $2k - 1$.
```

Table 1. The Scenario of Recursive Attack

In Step 5 of Table 1, because the attacker cannot confirm the same ECDBL signal in the vicinity of the top 5-th bit, he is able to decide this bit as -1. Hence, he measures the power signal C_2 about input point $21P = (2 * 11 - 1)P$. This power signal uses to know the upper 6-th bit, the next bit, of the secret key. For knowing the upper 6-th bit, the attacker guesses this bit as 1 and so he gets the power signal about input point $43P = (2 * 21 + 1)P$ in Step 4 of scenario. If this signal is compared with the signal C_2, it is as follows.

Input	1	1	$\bar{1}$	1	$\bar{1}$	**1**	$\bar{1}$	$\bar{1}$	1
$21P$	0 $21P$	$42P$ $63P$	$126P$ $105P$	$210P$ $231P$	$462P$ $441P$	$882P$ **903P**	**1806P** $1785P$	$\cdots$ $\cdots$	$\cdots$ $\cdots$
$43P$	0 $43P$	$86P$ $129P$	$258P$ $215P$	$430P$ $473P$	$946P$ **903P**	**1806P** $1849P$	$\cdots$ $\cdots$	$\cdots$ $\cdots$	$\cdots$ $\cdots$

Because the same ECDBL happens in the vicinity of the upper 6-th bit, the attacker can find the 6-th bit as 1. Through this mechanism, we can find the secret key d used in Algorithm 3 from MSB to LSB in sequence.

3.2 Initializing Attack

When recursive attack against sABS recoding recovers the secret key d from MSB to LSB in sequence, one bit of the secret key is exposed through inserting input value in the card at the minimum of one time and at the maximum of two times. We propose initializing attack as the second attack method against sABS recoding. This attack uses only one input value for recovering a bit of the secret key compared with recursive attack which uses 1.5 input value on the average. This method uses vulnerability that an attacker, with ease, can get an information of ECDBL from P to $2P$ for input value P in Step 4.1 of Algorithm 3.

Input value regulation Initializing attack, similar to recursive attack, is an attack method to break the next bit in sequence when it knows upper certain bits of the secret key d in advance. The basic idea of this attack is that an attacker selects an input value xP such that the intermediate value computed up to a guessing bit of the key always becomes P and so originates ECDBL operation from P to $2P$ at the computation time of the next bit. If we suppose that the attacker knows upper bits $t'_{n-1}t'_{n-2}...t'_{u+2}t'_{u+1}$ of sABS recoded value t' in Algorithm 3, let us consider him to anticipate t'_u as 1. If his guess is right, computed value of S until the moment of $i = u$, in Step 4 of Algorithm 3, about input value xP is $S = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u} + 1)xP$. Hence, he will select the value x such that $S = P$. If $k = \sum_{i=u+1}^{n-1} t'_i 2^{i-u-1}$, the value x is given by

$$x = (2k+1)^{-1} \; mod \; (\sharp E).$$

Here, $\sharp E$ represents order of elliptic curve in ECCs.

The order of an elliptic curve is in the form of $q,\ 2q,\ 4q,\ 6q(q : prime)$ in standard documents ANSI X9.62 [1], FIPS 186-2 [15], SECG [17], WTLS [19], and ISO/IEC 15946-4 [8]. Because $\gcd(2k+1, 6q) = 3$ is possible in spite of $\gcd(2k+1, 2) = 1$ and $\gcd(2k+1, q) = 1$, there is the case that $(2k+1)^{-1}$ does not exist in the case of order $6q$. But, in case of $2k+1 = 3t$, $gcd(2k-1, 6q) = 1$ is satisfied because $2k-1 = 3t-2$ is not multiple of 3. Accordingly, in case of $gcd(2k+1, \sharp E) \neq 1$, the attacker guesses that t'_u is -1 not 1. The computed value S up to $i = u$ is $S = (\sum_{i=u+1}^{n-1} t'_i 2^{i-u} - 1)xP$ when the input value is xP in the Step 4, and so the attacker selects value x such that $S = P$. If $k = \sum_{i=u+1}^{n-1} t'_i 2^{i-u-1}$, the value of x is given by

$$x = (2k-1)^{-1} \; mod \; (\sharp E).$$

Our attack seems to be Goubin's Refined Power-analysis Attack [6], but ours uses the discriminative method that compares two waveforms (The method is introduced in Section 4). Also, while Goubin's attack can use only "special point" with zero coordinate, our attack has a merit that can use almost every points over elliptic curve.

Scenario and Example of Initializing Attack In this section, we introduce the scenario of initializing attack for finding the entire value of the secret key, and then give a simple example to help understanding of this attack. Table 2 is the scenario of initializing attack for finding the entire value of the secret key.

Let us consider this scenario. For example, if the order of an elliptic curve is 73 and secret key d is $(101010011)_2$ in Algorithm 3, the value of t' becomes $11\bar{1}1\bar{1}1\bar{1}\bar{1}1$. In Table 2, suppose that the attacker already knows upper four bits of the secret key(upper 4 bits values : $11\bar{1}1 = (11)_{10}$). For now, he guesses the upper 5-th bit as 1 like Step 3 of scenario. Hence, the used input value to know this bit is $54P$ in Step 4 ($(2*11+1)^{-1}$ mod $73 = 54$). The attacker ascertains as the table below whether the card performs ECDBL of point P in the upper 5-th bit or not.

```
Step 1. Set $k = 1,\ i = 2$.
Step 2. If $i = n$, goto Step 8.
Step 3. If $(2k+1)^{-1}\ mod\ (\sharp E)$ exists, then $k' = 2k+1$, $s = 1$.
        Else then, $k' = 2k-1$, $s = -1$.
Step 4. Compute $k = k'^{-1}\ mod\ (\sharp E)$.
Step 5. Measure a power consumption $C$ related with the input point $kP$.
Step 6. If $C$ have an ECDBL signal from $P$ to $2P$ in the upper $i$-th bit, then $k = k'$.
        Else if $s = 1$, then $k = k' - 2$.
        Else then, $k = k' + 2$.
Step 7. $i = i + 1$, goto Step 2.
Step 8. If the output about the input point $P$ is $(2k+1)P$, then return $2k+1$.
        Else then, return $2k-1$.
```

Table 2. The Scenario of Initializing Attack

Input	1	1	$\bar{1}$	1	$\bar{\mathbf{1}}$	1	$\bar{1}$	$\bar{1}$	1
$54P$	0	$35P$	$67P$	$26P$	$14P$	**66P**	$\cdots$	$\cdots$	$\cdots$
	$54P$	$70P$	$13P$	$7P$	**33P**	$47P$	$\cdots$	$\cdots$	$\cdots$

In Step 6 of scenario, because the attacker cannot find ECDBL signal of point P in the upper 5-th bit, he is able to know this bit as -1 and so set $k = 23 - 2$. For knowing the upper 6-th bit recursively the attacker guesses this bit as 1, and so if he gets a power signal about input point $17P = (2 * 21 + 1)^{-1}P$ in Step 3 and Step 4, it is as follows.

Input	1	1	$\bar{1}$	1	$\bar{1}$	**1**	$\bar{1}$	$\bar{1}$	1
$17P$	0	$34P$	$29P$	$24P$	$9P$	$57P$	**2P**	$\cdots$	$\cdots$
	$17P$	$51P$	$12P$	$41P$	65	**P**	$58P$	$\cdots$	$\cdots$

Because the card performs ECDBL of point P in the upper 6-th bit, the attacker can know this bit as 1. Through this mechanism, attacker can find the entire value of the secret key d in sequence.

4 Statistical Approach of Noise Reduction

Both the proposed attacks and DA are accomplished if an attacker is able to become aware whether the smart card computes ECDBL of the same point or not through two ECDBL signals only. In this section, we propose that the above assumption can accomplish in actuality using experimental results. For example, we show that how to an attacker knowing the upper $(i-1)$ bits of the secret key detects the upper i-th bit in the recursive attack under assumption that he/she can distinguish power signals between ECDBL and ECADD.

Let $D_{i,1}^{(j)}$ $(D_{i,2}^{(j)})$ be a j-th ECDBL signal related with the first (second) input point $P_{i,1}$ $(P_{i,2})$ for knowing i-th bit. Also, $A_{i,1}^{(j)}$ $(A_{i,2}^{(j)})$ denotes a j-th ECADD signal related with the first (second) input point $P_{i,1}$ $(P_{i,2})$ for knowing i-th bit. Let the power signals related with input points $P_{i,1}$ and $P_{i,2}$ be represented as

$$P_{i,1} \Rightarrow D_{i,1}^{(1)} A_{i,1}^{(1)} D_{i,1}^{(2)} A_{i,1}^{(2)} D_{i,1}^{(3)} A_{i,1}^{(3)} D_{i,1}^{(4)} A_{i,1}^{(4)} D_{i,1}^{(5)} A_{i,1}^{(5)} \cdots ,$$
$$P_{i,2} \Rightarrow D_{i,2}^{(1)} A_{i,2}^{(1)} D_{i,2}^{(2)} A_{i,2}^{(2)} D_{i,2}^{(3)} A_{i,2}^{(3)} D_{i,2}^{(4)} A_{i,2}^{(4)} D_{i,2}^{(5)} A_{i,2}^{(5)} \cdots .$$

An attacker selects portions for $D_{i,1}^{(i)}$ and $D_{i,2}^{(i-1)}$, and then aligns two portions using 'alignment'. Experimental circumstance and setup are as follows:

Environment	PIC 16F84A microcontroller
Language	PIC programmer(Assembler)
Module	Scalar multiplication + sABS recoding + affine coordinate (Clock cycle of ECDBL: 3368)

First of all, for getting two distributions that is needed for judgement, we measure the three following power consumptions about time variable j.

- $S_1^{(i)}(j)$ related with the input point P_i.
- $S_2^{(i)}(j)$ related with the same input point P_i as before.
- $S_3^{(i)}(j)$ related with the different input point Q_i as before.

If the cryptographic device computes 160-bit scalar multiplications, $S_1^{(i)}(j)$ and $S_2^{(i)}(j)$ have 159 ECDBL signals for same point. Also, $S_1^{(i)}(j)$ and $S_3^{(i)}(j)$ have 159 ECDBL signals for different points. Before we define two distributions, the discriminant that can decide whether we have measured waveforms about the same operation or not is defined by

$$Disc.(S_1, S_2, t) = \frac{1}{m} \sum_{j=t+1}^{t+m} (S_1(j) - S_2(j))^2. \tag{6}$$

m is selected value in $[\lambda, n]$ where λ is the value including all coordinates x and y for the first time in ECDBL operation using an affine coordinate system and n is the length of signal related with 1 ECDBL.

Using this discriminant, we refer to two distributions X_1, X_2 as

$$X_1 = \bigcup_{i=1}^{L} \{Disc.(S_1^{(i)}, S_2^{(i)}, a) | a = k \times range, k \in \{0, 1, 2, ..., 158\}\},$$
$$X_2 = \bigcup_{i=1}^{L} \{Disc.(S_1^{(i)}, S_3^{(i)}, a) | a = k \times range, k \in \{0, 1, 2, ..., 158\}\}$$
$$where\ range\ is\ the\ length\ of\ signal\ related\ with\ 1\ ECDBL + 1\ ECADD$$
$$(range \approx \frac{approximate\ starting\ point\ of\ 159 - th\ ECDBL}{158}).$$

L is the value which decides the number of elements included in X_1, X_2.

In our research, sample rate is $100MS/s$, and so λ is about 60000 and n is about 336800 in equation (6). We select the value of m as 130000 in $[60000, 336800]$. Also, we select L as 3 in equation (7). Then, distributions X_1, X_2 are like a left side of Fig. 1. ($m_1 = E(X_1) = 24$, $a_1 = 63$, $m_2 = E(X_2) = 85$, $b_1 = 40$) where $E(\cdot)$ denotes the average of the distribution $\cdot$, a_1 denotes the maximum value of the distribution X_1, and b_1 denotes the minimum value of the distribution X_2.

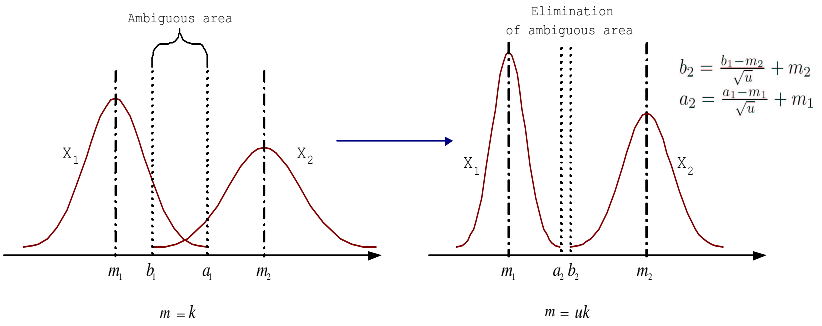
Ambiguous area
Elimination of ambiguous area
$b_2 = \frac{b_1 - m_2}{\sqrt{u}} + m_2$
$a_2 = \frac{a_1 - m_1}{\sqrt{u}} + m_1$
X_1
X_2
X_1
X_2
m_1
b_1
a_1
m_2
m_1
a_2 b_2
m_2
$m = k$
$m = uk$

Fig. 1. Distributions of ambiguous area and eliminated ambiguous area

An ambiguous area means the range that an attacker cannot decide whether signals related with the same operation or not. If the value of $Disc.(S_1, S_2, t)$ is in the ambiguous area ($40 \leq Disc.(S_1, S_2, t) \leq 63$ in ours), this value m for distinction must be selected in $[\lambda, n]$ to be the bigger value than the former. For reducing this error for each trial, the attacker must know the value of m that eliminates the ambiguous aria. In our experiment, we use the following proposition for eliminating this ambiguous area.

Proposition 1. *If $X_1 \leq a_1$, $X_2 \geq b_1$ are always completed with an error tolerance of $(\alpha/2)$ where $m = k$, the ambiguous area is eliminated where $m = (\frac{a_1 - b_1}{m_2 - m_1} + 1)^2 k$ with an error tolerance of $(\alpha - \alpha^2/4)$.*

Proof. According to the supposition, $P(X_1 \leq a_1) = P(X_2 \geq b_1) = \alpha/2$ where $m = k$.

If we convert this distributions into the standard normal distribution Z, the above equations are

$$P(Z \leq \frac{a_1 - m_1}{\sigma_1/\sqrt{k}}) = P(Z \geq \frac{b_1 - m_2}{\sigma_2/\sqrt{k}}) = \alpha/2 \ (\sigma_1^2 = var(X_1),\ \sigma_2^2 = var(X_2)).$$

If a_2, b_2 satisfy $P(X_1 \leq a_2) = P(X_2 \geq b_2) = \alpha/2$ where $m = uk$ (See the right side of Fig. 1.),

$$P(Z \leq \frac{a_2 - m_1}{\sigma_1/\sqrt{uk}}) = P(Z \geq \frac{b_2 - m_2}{\sigma_2/\sqrt{uk}}) = \alpha/2.$$

According to the above equations,

$$a_2 = \frac{a_1 - m_1}{\sqrt{u}} + m_1, \quad b_2 = \frac{b_1 - m_2}{\sqrt{u}} + m_2.$$

For eliminating the ambiguous area, the equation $b_2 > a_2$ must be satisfied, i.e.

$$u > (\frac{a_1 - b_1}{m_2 - m_1} + 1)^2.$$

The error tolerance (ET) is also $P(X_1 > a_2 \ or \ X_2 < b_2) = 1 - (1 - \alpha/2)^2 = (\alpha - \alpha^2/4)$. □

In the proposition, the value of α means the probability that the value of $Disc.(D_{i,1}^{(i)}, D_{i,2}^{(i-1)}, 0)$, in the practical attack, escapes previously measured bounds. This value depends on how many experiments have been carried out previously. If we compute the maximum ET when the ambiguous area is eliminated, these values are as follows according to the frequency L of the experiment.

L	1	2	3	4	5
ET	3.142×10^{-3}	1.572×10^{-3}	1.048×10^{-3}	7.860×10^{-4}	6.288×10^{-4}

In our preliminary research, distributions X_1 and X_2 about *Disc.*s can have 477 values for each, and so $P(X_1 > a_1) = P(X_2 < b_1)$ is less than 1/477 approximately. For this reason, because α is less than 1/954, the ET is also less than 1.048×10^{-3}. By proposition, we used m as 246514 and the discriminating value of *Disc.* as 52.321 ($= a_2 = b_2$). In other words, if the *Disc.* value of compared two ECDBL signals ($D_{i,1}^{(i)}$ and $D_{i,2}^{(i-1)}$ in recursive attack) is greater than 52.321, ECDBLs about different points have been carried out; otherwise, ECDBLs about same point have been carried out. Using these selected values through the preliminary research, we practically find the secret key comparing two ECDBL signals.

5 Countermeasures against proposed attacks

In this section, we consider an environment that our attacks are applicable and a countermeasure to resist our attacks. First of all, our attacks and original DA can only be carried out in the affine coordinate. Suppose that the smart card uses a projective coordinate system. And then, even if ECDBL operations about the same point on an elliptic curve are carried out, values of each coordinate may not be different like Fig. 2. Hence, because values of coordinate are different, ECDBL

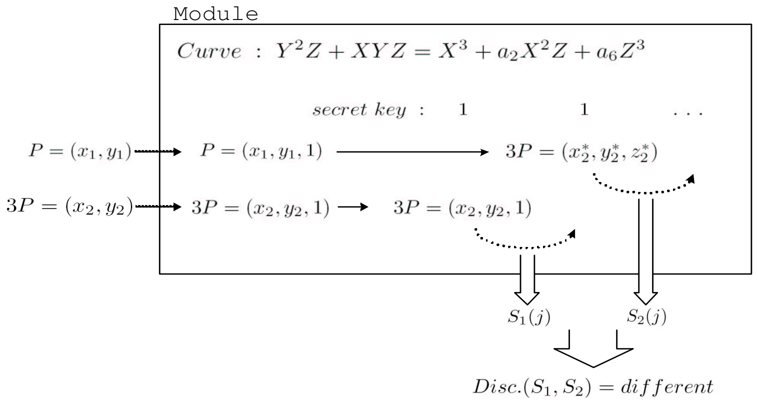

Fig. 2. The attack in projective coordinate system

signals for the same point could be considered as ECDBL about different points from the viewpoint of the attacker.

And now, we consider a countermeasure on proposed attacks. Because those use the method that chooses input points corresponding to the guessed bit of a recoded secret key, those can break recoding methods of the secret key for SPA countermeasures. Hence, the countermeasure against our attacks must use random point. Various countermeasures using random point are introduced so far. Among them, BRIP [13] proposed by Mamiya et al. can be applied to our attacks efficiently.

6 Conclusion

In this paper we have proposed two attacks, recursive attack and initializing attack, against sABS countermeasure proposed by Hedabou et al. As these analyses classified into SPA the method that extend the DA, those enlarge the range of attack. We have performed an experiment to justify the possibility of our attacks. The concrete method of this experiment and the backing of the proposition can furnish an practical information about these analysis methods.

Acknowledgements

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Advancement)" (IITA-2006-(C1090-0603-0025)). We would like to thank the anonymous referees for various comments and improvements.

References

1. ANSI X9.62, Public Key Cryptography for the Financial Services Industry, *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 1999.
2. E. Biham and A. Shamir, *Differential Fault Analysis of Secret Key Cryptosystems*, CRYPTO 1997, LNCS 1294, pp. 513-525, 1997.
3. C. Clavier, M. Joye, *Universal Exponentiation Algorithm . A First Step towards Provable SPA-Resistance*, CHES 2001, LNCS 2162, pp. 300-308, 2001.
4. J.S. Coron, *Resistance against differential power analysis for Elliptic Curve Cryptosystems*, CHES 1999, LNCS 1717, pp. 292-302, 1999.
5. P.A. Fouque and F. Valette, *The Doubling Attack. Why Upwards Is Better than Downwards*, CHES 2003, LNCS 2779, pp. 269-280, 2003.
6. L. Goubin. *A refined power analysis attack on elliptic curve cryptosystems*, PKC 2003, LNCS 2567, pp. 199-211, 2003.
7. M.Hedabou, P.Pinel, and L. Bebeteau, *Countermeasures for Preventing Comb Method Against SCA Attacks*, ISPEC 2005, LNCS 3439, pp. 85-96, 2005.
8. ISO/IEC 15946-4, *Information technology - Security techniques . Cryptographic techniques based on elliptic curves - Part 4: Digital signatures giving message recovery.* Working Draft, JTC 1/SC 27, December 28th, 2001.
9. N. Koblitz, *Elliptic curve crypto- systems*, Math. of Computation, Vol.48, pp. 203-209, 1987.
10. P. Kocher, J. Jaffe, and B. Jun, *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS,and Others Systems.* CRYPTO 1996, LNCS 1109, pp. 104-113, 1996.
11. P. Kocher, J. Jaffe, and B. Jun, *Introduction to differential power analysis and related attacks*, http://www.cryptography.com/dpa/technical, 1998.
12. P. Kocher, J. Jaffe, and B. Jun, *Differential power analysis*, CRYPTO 1999, LNCS 1666, pp. 388-397, 1999.
13. H. Mamiya, A. Miyaji, H. Morimoto *Efficient Countermeasures Against RPA, DPA, and SPA*, CHES 2004, LNCS 3156, pp. 343-356, 2004.
14. Victor S. Miller, *Use of Elliptic Curves in Cryptography*, CRYPTO 1985, LNCS 218, pp. 417-426, 1985.
15. National Institute of Standards and Technology (NIST), *Recommended Elliptic Curves for Federal Government Use.* In the appendix of FIPS 186-2, available from http://csrc.nist.gov/publications/fips/fips186-2/fips186-2.pdf
16. R.L. Rivest, A. Shamir, and L.M. Adleman. *A method for obtaining digital signatures and public-key cryptosystem.* Communications of the ACM, 21(2):120-126, 1978.
17. Standards for Efficient Cryptography Group (SECG), *Specification of Standards for Efficient Cryptography*, Ver. 1.0, 2000. Available from http://www.secg.org/secg docs.htm
18. C.C. Tiu, *A New Frequency-Based Side Channel Attack forEmbedded Systems*, master's thesis, University of Waterloo, 2005.
19. Wireless Application Protocol (WAP) Forum, *Wireless Transport Layer Security (WTLS) Specification.* Available from http://www.wapforum.org
20. S.M. Yen, S.J. Kim, S.G. Lim, and S. J. Moon, *A countermeasure against one physical cryptanalysis May Benefit Another Attack*, ICISC 2001, Korea. Dec. 2001.