Generic Cryptanalysis of Combined Countermeasures with Randomized BSD Representations

Tae Hyun Kim[1], Dong-Guk Han[2], Katsuyuki Okeya[3], and Jongin Lim[1]

[1] Center for Information and Security Technologies(CIST),
Korea University, Seoul, KOREA
{thkim,jilim}@cist.korea.ac.kr
[2] FUTURE UNIVERSITY-HAKODATE, 116-2 Kamedanakano-cho, Hakodate,
Hokkaido, 041-8655, Japan
christa@fun.ac.jp
[3] Hitachi, Ltd., Systems Development Laboratory,
1099, Ohzenji, Asao-ku, Kawasaki, 215-0013 Japan
ka-okeya@sdl.hitachi.co.jp

Abstract. In ICICS'04, Sim et al. proposed an attack against the full version of Ha-Moon's countermeasure which is one of enhanced countermeasures. The analysis technique is based on the fact that the probability for the appearance of an intermediate value is $p = 1/2$. By our simulations, however, it is proven to be not true. Thus sometimes the output of their attack might be wrong because there exists the case that the probability p is so small that they can make a wrong decision.
In this paper we repair the above attack, and then propose a generic analytical technique applicable to all BSD type countermeasures combined with some simple power analysis countermeasures. In order to show that the proposed attack is as practical as the usual differential power analysis (DPA), we estimate the number of samples and computational cost. Furthermore, we enhance the proposed attack in two ways such that it works against right-to-left algorithm in a simpler and more efficient way, and also works against one combined with an extra DPA countermeasure.

Keywords: *Elliptic Curve Cryptosystems, Side Channel Attack, Differential Power Analysis, Refined Power Analysis, Binary Signed Digit (BSD) Representation.*

1 Introduction

Mobile devices such as smart cards, mobile phones, and handheld computers are penetrating in our daily life in order for us to be convenient. Since mobile devices are equipped with scarce resources only, cryptographic algorithms on them should be optimized. Above all, elliptic curve cryptosystems (ECC) [13, 17] are suitable for implementing on such devices because of the reduced key size required in comparison to other cryptosystems (e.g. a 160-bit ECC has almost the same security as a 1024-bit RSA).

On the other hand, side channel attacks (SCA) have been recognized as menaces to ECC. In SCA, an attacker observes side channel information such as computation timing, power consumption, and electro-magnetic radiation while a cryptographic device performs cryptographic operations, then analyzes the information for revealing the secret stored in the device [12]. Thus constructing an efficient scalar multiplication method which is secure against SCA and analyzing its security are important research topics [3, 10, 26].

For this purpose, many countermeasures against SCA have been proposed. In particular, a popular type of countermeasures is based on inserting random decisions when choosing one representation among several different representations for the same secret scalar. For instance, it includes Oswald-Aigner countermeasure [20], Ha-Moon countermeasure [8], Ebeid-Hasan countermeasure [5], and the countermeasure of Agagliate et al. [1], which are based on randomized Binary Signed Digit (BSD) representations. Moreover, this type of countermeasures on ECC provides us with good performance/efficiency. We call countermeasures using BSD representations *BSD type countermeasure*[4].

Whereas many BSD type countermeasures were proposed, most of them have been broken by many sophisticated simple power analysis (SPA) if we use them as a single countermeasure against SCA [23, 14, 21, 9, 25].

A possible approach to resist the sophisticated SPAs is to combine BSD type countermeasures with an SPA countermeasure using a fixed procedure such as Coron's dummy method [4] or Montgomery ladder methods [19, 22]. An example of BSD type combined with an SPA countermeasure is the full version of Ha-Moon's method [8] composed of a random recoding method and an SPA-immune algorithm using dummy operations. Unfortunately, the full version of Ha-Moon's method has been analyzed by two different methods [6, 24]. The attacks utilize a characteristic of BSD representations generated by a specific random recoding method. Thus, the attacks are ad-hoc in the sense that it is tailored specifically to Ha-Moon's countermeasure. If the target countermeasure is changed from Ha-Moon's countermeasure then the characteristic is also changed. Thus, it is not clear whether the attacks can be applicable to the other BSD type countermeasures or not.

1.1 Contributions of This Paper

The proposed attack can break the combined countermeasures without knowledge for the appearance probability of an intermediate point in advance, i.e., it is independent of a random recoding method. Therefore, the proposed attack is applicable to not only Ha-Moon's countermeasure but also *all BSD type countermeasures* under reasonable assumptions. Moreover, to reduce the unwanted noise in power signals, we use a model of the signal-to-noise ratio (SNR) for "Zero Exponent Multiple Data" (ZEMD) [15]. In this model, the role of the

[4] The BSD representations use the set of digits $\{-1, 0, 1\}$. Thus, in this paper, we do not deal with countermeasures based on window methods using randomized addition chains.

number of samples used in the ZEMD attack is very important. In this paper, we show how many number of samples are required to obtain the same height of peaks as the ordinary ZEMD attack on unprotected algorithms. From our simulations, we deduce that the proposed attack is as practical as the ordinary ZEMD attack.

In this paper, we propose analysis techniques against the following three targets.

Target 1. The BSD type countermeasures combined with an SPA countermeasure using a fixed procedure such as Coron's dummy method [4] or Montgomery ladder methods [19, 22]. An example of SPA countermeasure is the following Addition-Subtraction_Always method.

Addition-Subtraction_Always method

INPUT A point P, and $k = \sum_{j=0}^{n-1} k_j 2^j$, $k_j \in \{0, 1\}$
$d = \sum_{j=0}^{n} d_j 2^j$, $d_j \in \{-1, 0, 1\}$, where d is a recoded number of k
OUTPUT $Q = dP$

1. $Q[0] \leftarrow \mathcal{O}$, $R[0] \leftarrow P$, $R[1] \leftarrow P$, $R[2] \leftarrow -P$
2. for $j = n$ downto 0
   2.1. $Q[0] \leftarrow$ECDBL$(Q[0])$
   2.2. $Q[1] \leftarrow$ECADD$(Q[0], R[1 - d_j])$
   2.3. $Q[0] \leftarrow Q[|d_j|]$
3. Return $Q[0]$

Target 2. The BSD type countermeasures combined with a DPA countermeasure using randomized point representation methods such as Coron's third method called randomized projective coordinates [4] and random isomorphism methods [11].

Target 3. The BSD type countermeasures using right-to-left computations.

This paper is organized as follows. In the next section, we introduce some tools for power analysis. In Section 3 we propose a generic attack against Target 1 and show simulation results. In Section 4, we enhance the proposed attack in two ways: against Target 2 and Target 3. In Section 5 we show a comparison of attacks against BSD type countermeasures. Finally, we conclude in Section 6.

2 Tools for Power analysis

To construct an attack against BSD type combined with an SPA countermeasure, we introduce two concepts; discernment point and signal-to-noise ratio.

2.1 Discernment Point in ZEMD Attack

In this subsection, we introduce the concept of *discernment point*. A ZEMD attack utilizes a correlation between power consumption and any specific key-dependent bits. In view of ZEMD attack, the three following assumptions support the success of ZEMD attack.

(i) A point that its appearance in the middle of computing provides the attacker with information of (a portion of) the secret key exists. Such a point is referred to as *discernment point*.
(ii) A coordinate of the discernment point is computable or predictable with purposive probability for the attacker.
(iii) The attacker can discern whether the discernment point appears or not using side channel information.

The attacker succeeds in ZEMD attack under the three assumptions.

(1) The attacker classifies input points into two classes depending on the coordinate of the discernment point. ((i) provides the existence of a discernment point, and (ii) provides the attacker's capability of classifying.)
(2) The attacker collects side channel information, and discerns whether the discernment point appears or not. ((iii) provides the attacker's capability of discerning.)
(3) The attacker reveals a portion of the secret key using the (dis)appearance of the discernment point. ((i) provides the attacker's capability of revealing.)

Now, we simply describe a ZEMD attack against SPA-protected Double-Add_Always method by using the above concept of the discernment point.

Double-Add_Always method

INPUT A point P, and $k = \sum_{j=0}^{n-1} k_j 2^j$, $k_j \in \{0, 1\}$
OUTPUT $Q = kP$

1. $Q[0] \leftarrow P$
2. for $j = n - 2$ downto 0
   2.1. $Q[0] \leftarrow$ECDBL$(Q[0])$
   2.2. $Q[1] \leftarrow$ECADD$(Q[0], P)$
   2.3. $Q[0] \leftarrow Q[k_i]$
3. Return $Q[0]$

An intermediate point which is actually calculated at the step 2.1 after $j = i$ bit (k_i) calculation in Double-Add_Always method is as follows;

– $(\sum_{j=i+1}^{n-1} k_j 2^{j-i+1}) \cdot P$ if $k_i = 0$, and $(\sum_{j=i+1}^{n-1} k_j 2^{j-i+1} + 2) \cdot P$, if $k_i = 1$.

Thus the discernment point used in the ZEMD attack on Double-Add_Always method can be one of $(\sum_{j=i+1}^{n-1} k_j 2^{j-i+1}) \cdot P$ and $(\sum_{j=i+1}^{n-1} k_j 2^{j-i+1} + 2) \cdot P$.

2.2 Signal-to-Noise Ratio

We introduce the concept of signal-to-noise ratio (SNR) in order to estimate the required number of samples. A successful ZEMD attack requires that an attacker can detect the signal over the noise. To reduce the unwanted noise in the power signal, Messerges et al. used filtering strategies [16]. They proposed a model for the ZEMD signal-to-noise ratio.

Proposition 1 ([16]). *A ZEMD attack using R samples on an M-bit processor in* Double-Add_Always method*, with signal size ε, average nonalgorithmic noise variance σ^2, and percentage of algorithmic noise α, has a voltage intrasignal SNR that can be modeled by*

$$SNR = \frac{\varepsilon\sqrt{R}}{\sqrt{8\sigma^2 + \varepsilon^2(\alpha M + M - 1)}}. \quad (1)$$

3 Proposed Attack

In this section, we propose a novel ZEMD attack algorithm against Target 1 and show the results of simulation. (The proofs of several propositions may be found in appendix.)

3.1 Notations

Let $k = \sum_{j=0}^{n-1} k_j 2^j$ with $k_j \in \{0, 1\}$ be the n-bit binary secret key and $d = \sum_{j=0}^{n} d_j 2^j$ with $d_j \in \{-1, 0, 1\}$ be the $(n+1)$-bit random recoded number generated from k by a random recoding method. Note that k and d are obviously the same number, even though their representations differ. Let $k_{[s_1,t_1]}$ and $d_{[s_1,t_1]}$ denote partial bits from the s_1-th bit to the t_1-th bit of k and d, respectively. Namely $k_{[s_1,t_1]} := \sum_{j=s_1}^{t_1} k_j 2^{j-s_1}$ and $d_{[s_2,t_2]} := \sum_{j=s_2}^{t_2} d_j 2^{j-s_2}$. Here, $0 \le s_1 \le t_1 \le n-1$ and $0 \le s_2 \le t_2 \le n$.

- R: The number of executions that an attacker observes, i.e., the number of samples.
- R^S: The smallest number of executions to detect the signal over the noise in SNR formula (1) against Double_Add_Always method. If R is chosen as $R \ge R^{\mathcal{S}}$ then an attacker can break Double_Add_Always method.
- $R^{\mathcal{M}}$: The maximum number of execution which an attacker can use in a ZEMD attack. Note that $R^{\mathcal{M}}$ depends on the computational power of an attacker. By the definition, $R^{\mathcal{M}} \ge R$.
- P_r: For $1 \le r \le R$, the r-th input point into Addition-Subtraction_Always method.
- IP_r^i: The intermediate point which is actually calculated at the step 2.1 after $j = i$ bit (k_i) calculation for the r-th input point into Addition-Subtraction _Always method.
- p: The appearance probability of a discernment point after the calculation for k_i, i.e., $p := Pr[DP_r^i = IP_r^i]^{1 \le r \le R}$ during R executions. Note that the probability depends on the secret key and a random recoding method used in BSD type countermeasures.

3.2 SNR for Probabilistic Appearance of Discernment Point

As Addition-Subtraction_Always method uses randomized BSD representations, the size of signals may decrease due to probabilistic appearance of discernment

points. Thus the SNR model for BSD type countermeasure should be modified as follows:

Proposition 2. *Assume that the computational environment is the same as Proposition 1. If the appearance probability of a discernment point is p, then the signal-to-noise ratio is*

$$SNR = \frac{\varepsilon p\sqrt{R}}{\sqrt{8\sigma^2 + \varepsilon^2(\alpha M + M - p)}}. \tag{2}$$

Thus the SNR is approximately p times larger than the original. In other words, in order to obtain the same SNR as the original, the required samples are p^{-2} times larger than the original.

Remark 1. Since the required number of samples R is determined by the probability p, if $p^{-2}R$ is bigger than $R^{\mathcal{M}}$ or less than $R^{\mathcal{S}}$, then he/she may not obtain a useful signal over the noise even if the attacker's guess was right.

3.3 Properties of all BSD type Countermeasures

In all BSD type countermeasures we justify that the following property is satisfied, i.e. the consequence of the following proposition does not depend on the choice of a recoding technique.

Proposition 3. *$d_{[i,n]}$ is either $k_{[i,n-1]}$ or $k_{[i,n-1]} + 1$.*

From Proposition 3, we can obtain a relation between the i-th bit k_i of secret key and intermediate point (IP_r^i).

Observation 1.

$$\begin{array}{llll} IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]}\right) \cdot P_r & \text{or} & \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 1\right) \cdot P_r, & \text{if } k_i = 0; \\ IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 1\right) \cdot P_r & \text{or} & \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 2\right) \cdot P_r, & \text{if } k_i = 1. \end{array}$$

Thus we can see that there are three kinds of intermediate points. The intermediate point $IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]}\right) \cdot P_r$ only appears in the case of $k_i = 0$, and $IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 2\right) \cdot P_r$ only appears in the case of $k_i = 1$. But $IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 1\right) \cdot P_r$ is related to both $k_i = 0$ and $k_i = 1$.

The above observation helps us to determine a discernment point DP_r^i to recover k_i. We concretely describe how to determine discernment points in the next section.

3.4 Proposed Attack

Before formally describing our analytical framework, we will make some assumptions more precisely. Our analysis depends on the following assumptions:

Assumption 1. (1) We assume that the scalar multiplication in Target 1 utilizes a *left-to-right computation*, i.e., the secret key is scanned from the most significant bit.
(2) We can repeatedly obtain the measurement of power consumption at the device for the fixed secret key k.
(3) Suppose an attacker already knows the highest bits $k_{n-1}, \ldots, k_{i+1}$ of the secret key k. The attacker will try to recover the next bit k_i with the ordinary ZEMD attack. Assume the attacker first uses $\left(2^2 \cdot k_{[i+1,n-1]}\right) \cdot P_r$ as the discernment point to check $k_i = 0$.

The second assumption is reasonable. For some elliptic curve schemes, collecting power signals for the fixed secret key may be impossible, like the signature generation of ECDSA. However, some other schemes like ECDH are possible. We can then obtain the following result under Assumption 1.

Proposition 4. *If the appearance probability of the discernment point for the target bit k_i is p ($\neq 0$), the use of $p^{-2}R^{\mathcal{S}}$ samples enables the attacker to recover k_i; $k_i = 0$ if an appreciable peak occurs, or $k_i = 1$ if not.*

By Proposition 4, if the attacker uses R samples such that $R \geq p^{-2}R^{\mathcal{S}}$ then he/she can recover k_i. But, there are two cases that the attacker can not find any appreciable peak over noise in the ordinary ZEMD attack, even though his/her guess is right.

Problem 1: The case of the probability $p = 0$. Namely, $IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 1\right) \cdot P_r$ always occurs during R executions. Thus the attacker may confuse whether k_i is 0 or not.
Problem 2: The case that the probability p is so small such that $R = p^{-2}R^{\mathcal{S}} > R^{\mathcal{M}}$. It implies the attacker can't use $R = p^{-2}R^{\mathcal{S}}$ samples to determine k_i.

Furthermore, there is one more problem that it is difficult for the attacker to predict the probability p in advance because the probability depends on the secret key and the used recoding method. Thus he/she can not determine the exact number of sample R such that $R \geq p^{-2}R^{\mathcal{S}}$.

Remark 2. In [24], Sim et al. assumed that the appearance probability of an intermediate point is always 1/2 (i.e., $p = 1/2$) because of a random bit. So, they mentioned that the required number of samples should be doubled in order to detect the same height of peaks as that of the ordinary ZEMD attack on unprotected scalar multiplication algorithms. Unfortunately, the assumption is not always true. Actually, the probability depends on both the used random recoding method and the secret scalar. Thus, their attack is not practical in the sense of the number of samples.

We now describe how to solve these problems. Let us assume that we always use the maximum number of samples $R^{\mathcal{M}}$ to recover k_i, i.e. $R = R^{\mathcal{M}}$. Then the smallest probability p that we can recognize appreciable peaks is $\sqrt{\frac{R^{\mathcal{S}}}{R^{\mathcal{M}}}}$

by Proposition 2. Let the smallest appearance probability $p = \sqrt{\frac{R^{\mathcal{S}}}{R^{\mathcal{M}}}}$ that an attacker can detect peaks be denoted as $\mathcal{LB}$, i.e., it means a lower bound of the appearance probability. In other words, $p \geq \mathcal{LB}$ is equivalent to $R^{\mathcal{M}} \geq p^{-2}R^{\mathcal{S}}$, that is, the number of used samples $R^{\mathcal{M}}$ is enough to detect some useful peak in the obtained power consumption signal. (Note that $\mathcal{LB}$ depends on the ability of an attacker because the capability to obtain $R^{\mathcal{M}}$ differs each.) From Observation 1, we can construct a new attack strategy as follows.

The attack method by one bit guess:

Assumption: We always use $R^{\mathcal{M}}$ samples, i.e., $R = R^{\mathcal{M}}$.
Step 1: Use the discernment point $DP_r^i = \left(2^2 \cdot k_{[i+1,n-1]}\right) \cdot P_r$. If some useful peaks appear over noise in SNR, i.e., $p \geq \mathcal{LB}$, then output $k_i = 0$
Step 2: Else, use another $DP_r^i = \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 2\right) \cdot P_r$. If some useful peaks appear over noise in SNR, i.e., $p \geq \mathcal{LB}$, then output $k_i = 1$.
Step 3: Otherwise, we can not determine whether k_i is 0 or not because $IP_r^i = \left(2^2 \cdot k_{[i+1,n-1]} + 2 \cdot 1\right) \cdot P_r$ is operated with high probability.

We now solve the case that k_i is not determined in Step 3. Assume that we guess more bits instead of one bit in the above attack. For simplicity, we explain the case of two bits guess. For all $(k_i k_{i-1})_2$, the intermediate point IP_r^{i-1} is as follows:

Observation 2.

$IP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]}) \cdot P_r$ or $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 1) \cdot P_r$, if $(k_i k_{i-1})_2 = (00)_2$;
$IP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 1) \cdot P_r$ or $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 2) \cdot P_r$, if $(k_i k_{i-1})_2 = (01)_2$;
$IP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 2) \cdot P_r$ or $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 3) \cdot P_r$, if $(k_i k_{i-1})_2 = (10)_2$;
$IP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 3) \cdot P_r$ or $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 4) \cdot P_r$, if $(k_i k_{i-1})_2 = (11)_2$.

From the above observation, we can find some useful relations.

- $IP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]}) \cdot P_r$ or $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 4) \cdot P_r$ only appears in the case of $(k_i k_{i-1})_2 = (00)_2$ or $(11)_2$, respectively.
- If $IP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 1) \cdot P_r$ or $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 3) \cdot P_r$ appears, then $k_i = 0$ or 1, respectively.

There is one more good relation; for example, if the appearance probabilities for both $DP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 1) \cdot P_r$ and $(2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 2) \cdot P_r$ satisfy the condition i.e. $p \geq \mathcal{LB}$, then we can be convinced that $(k_i k_{i-1})_2 = (01)_2$. More exactly, suppose an attacker uses $R^{\mathcal{M}} = 9R^{\mathcal{S}}$, i.e., $\mathcal{LB} = 1/3$, and $p = 0.6$ when $DP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 1) \cdot P_r$ and $p = 0.4$ when $DP_r^{i-1} = (2^3 \cdot k_{[i+1,n-1]} + 2 \cdot 2) \cdot P_r$. Then he/she can detect $(k_i k_{i-1})_2 = (01)_2$ because these two probabilities p, 0.6 and 0.4, are greater than $\mathcal{LB} = 1/3$. By recursively processing the above way, we can recover the remaining bits.

3.5 Simulations

In this subsection, we estimate the maximum number of consecutive guess bits and the number of trial guess required in the proposed attack by computing the

Table 1. For a 160-bit secret key, the number of trial guess in the proposed ZEMD attack against BSD type countermeasures with SPA countermeasure depending on the ability to obtain $R^{\mathcal{M}}$

Countermeasures	$R^{\mathcal{M}}$					
	$2R^S$	$5R^S$	$10R^S$	$20R^S$	$50R^S$	$100R^S$
Ha-Moon [8]	16484	2716	1987	1480	1019	868
Ebeid-Hasan [5]	1515	711	596	543	508	478
Agagliate et al. [1]	4004	2917	2915	2898	2896	2893

appearance probability of intermediate points in software written in C-language. We carried out simulations on the proposed attack applied to Ha-Moon's method, Ebeid-Hasan's method, and that of Agagliate et al. as follows.

1. Implement Ha-Moon's method, Ebeid-Hasan's method, and that of Agagliate et al. with Addition-Subtraction_Always method on typical microprocessors: Pentium IV/2GHz (32-bit μP; Windows XP, MSVC).
2. For $i = 1$ to $10,000$ do
   - Choose a 160-bit scalar randomly.
   - Obtain 10,000 random recoded numbers generated from the secret key by random recoding methods.
   - Compute the appearance probability of two intermediate points for all bits using the given 10,000 random recoded numbers, that is, $Pr\big[d_{[i,n]} = k_{[i,n-1]}\big]$ and $Pr\big[d_{[i,n]} = k_{[i,n-1]} + 1\big]$.
   - For each $\mathcal{LB}$, we recover the secret key by the proposed attack.
   - From the result of the above step, count the maximum number of consecutive guess bits and the number of trial guess.
3. Compute the average of the maximum number of consecutive guess bits and the number of trial guess for each $\mathcal{LB}$. (i.e. sum of the maximum number of consecutive guess bits / 10,000 and sum of the number of trial guess /10,000)

Table 1 shows the number of trial guess in the proposed ZEMD attack against BSD type countermeasures with Addition-Subtraction_Always method depending on the ability to obtain $R^{\mathcal{M}}$. The number of trial guess means the number of discernment points used in ZEMD attack. Note that since the original ZEMD attack requires one discernment point to recover one bit, the original ZEMD attack requires n times trial guess for n-bit secret key.

Fig. 1 shows the required number of measurements to reveal the secret key depending on the computational power of the attacker (i.e. $\mathcal{LB}$). If an attacker has the capability of guessing consecutive 10 bits then, to obtain the same SNR as Proposition 1, the required samples are about $2R^{\mathcal{S}}$ in Ha-Moon's countermeasure, $1.4R^{\mathcal{S}}$ in Ebeid-Hasan's countermeasures, and $R^{\mathcal{S}}$ in Agagliate et al.'s countermeasure. In the case of Agagliate et al.'s countermeasure, we can recover the secret key with the same number of samples as the ordinary ZEMD attack.

From Fig. 1 and Table 1, we can derive the following conclusion. If we use $10R^{\mathcal{S}}$ samples then $\mathcal{LB} \approx 0.32$. Thus we can determine the whole 160-bits secret

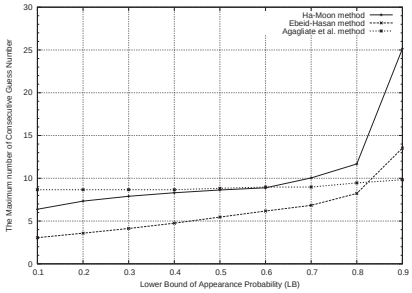

Fig. 1. Relationship between the number of samples $R^{\mathcal{M}}$ and the maximum number of consecutive guess bits for a 160-bits secret key

key using 1987 trial guess before at most 10-bits consecutive guess in Ha-Moon's method, 596 trial guess before at most 7-bits consecutive guess in Ebeid-Hasan's method, and 2915 trial guess before at most 9-bits consecutive guess in Agagliate et al.'s method. From these observations, we can see that the proposed ZEMD is as practical as the ordinary ZEMD attack.

4 Enhancing the Proposed Attack

The proposed attack can be enhanced in two ways: (1) against *right-to-left* computation. (2) against BSD type countermeasure combined with a *DPA countermeasure*.

4.1 Attack for Right-to-Left Algorithm

The proposed attack is also applicable to *right-to-left computation*. In order to construct an attack against a right-to-left algorithm, we use the following property:

Proposition 5. *$d_{[0,i]}$ is either $k_{[0,i]}$ or $k_{[0,i]} - 2^{i+1}$.*

Proposition 5 is easily derived from Proposition 3. We obtain the relation between k_i and intermediate point IP_r^i. Here, IP_r^i denotes the intermediate point which is actually calculated at ECADD after $j = i$ bit (k_i) calculation for the r-th execution in a right-to-left version of Addition-Subtraction_Always method.

Observation 3.

$$\begin{array}{lll} IP_r^i = \left(k_{[0,i-1]} + 2^{i+1}\right) \cdot P_r & \text{or} \quad k_{[0,i-1]} \cdot P_r, & \text{if } k_i = 0; \\ IP_r^i = \left(k_{[0,i-1]} + 2^i + 2^{i+1}\right) \cdot P_r & \text{or} \quad \left(k_{[0,i-1]} + 2^i\right) \cdot P_r, & \text{if } k_i = 1. \end{array}$$

By above observation, the intermediate points for $k_i = 0$ are totally different from those for $k_i = 1$. Thus the proposed attack against right-to-left algorithms can use both cases of intermediate points for k_i as the discernment point, e.g. the discernment point could be $(k_{[0,i-1]} + 2^{i+1}) \cdot P_r$ or $k_{[0,i-1]} \cdot P_r$ to check $k_i = 0$.

Proposition 6. *In Right-to-Left algorithm, the number of try to detect secret key bit k_i is at most two times. As there is no collision between intermediate points for $k_i = 0$ and intermediate points for $k_i = 1$, first we use $IP_r^i = (k_{[0,i-1]} + 2^{i+1}) \cdot P_r$, if we detect useful peaks over noise with SNR then k_i is 0. Otherwise, we try again with $IP_r^i = k_{[0,i-1]} \cdot P_r$. If we find useful peaks over noise with SNR then k_i is 0, otherwise k_i is 1.*

Therefore, the proposed attack against right-to-left algorithms can recover the secret key bit by bit as the similar to the ordinary ZEMD attack. Thus the proposed attack against a right-to-left algorithm is more simple and efficient than that against a left-to-right one.

Remark 3. Since Oswald-Aigner's method [20] is a right-to-left algorithm, it is very easily broken using the proposed attack against right-to-left algorithms.

4.2 RPA Attack

In this section, we discuss the security of BSD type countermeasures combined with a DPA countermeasure using randomized point representation methods.

In order to strengthen the security of BSD type countermeasures, the BSD type may further be combined with some DPA countermeasures using randomized point representation methods such as randomized projective coordinates [4] or random isomorphisms method [11] before scalar multiplications.

However, Goubin proposed the refined power analysis (RPA) using "special point" $(x, 0)$ and $(0, y)$ that cannot be randomized by randomized point representation techniques [7]. Thus the proposed attack can also break BSD type countermeasures combined with randomized point representation methods by using the "special point" as a discernment point.

Note that other notations and assumptions are the same as those in the previous sections with the exception of combining Addition-Subtraction_Always method with DPA countermeasure. We can then find two differences between DPA and RPA as follows:

SNR of RPA in the case using BSD type countermeasures: Similar to DPA on Addition-Subtraction_Always method described in Proposition 1 and 2, we propose a proposition which deals with SNR of RPA on Addition-Subtraction_Always method with the DPA countermeasure using randomized point representation.

Proposition 7. *Assume that the computational environment is the same as Proposition 1. If the appearance probability of the "special" point P_0 is p', then the signal-to-noise ratio is*

$$SNR = \frac{\varepsilon p' M \sqrt{R}}{\sqrt{8\sigma^2 + \varepsilon^2(\alpha M + M - Mp')}}. \tag{3}$$

Thus the SNR is approximately p' times larger than the original RPA. In other words, in order to obtain the same SNR as the original, the required samples are p'^{-2} times larger than the original.

Note that the proof of it is similar to that of Proposition 2 and refer to the Theorem 3 in [16]. Actually, the appearance probability p of the discernment point in the proposed ZEMD attack is exactly the same as the appearance probability p' of the "special" point P_0.

Remark 4. Proposition 7 shows that an attacker requires approximately $M^{-2}R$ samples to obtain the same SNR as Proposition 2 (for the proposed ZEMD attack).

Adaptively Chosen Data Attack: RPA is an adaptively chosen data attack. Since RPA requires the special point for detecting a specific bit of the scalar, the observed samples cannot be reused. That is, for detecting each bit, the attacker has to observe power consumptions for new data. So, in the proposed RPA attack, the maximum number of samples that the attacker can use for each trial guess is $R^{\mathcal{M}}/$*the number of trial guess* on average. Thus, the symbol $\mathcal{LB} = \sqrt{R^{\mathcal{S}}/R^{\mathcal{M}}}$ used in the new attack strategy in section 3.4 is replaced with $\sqrt{(R^{\mathcal{S}} \times \textit{the number of trial guess})/R^{\mathcal{M}}}$. For a more successful attack we can use more smaller samples if $R^{\mathcal{M}}/\textit{the number of trial guess} > R^{\mathcal{M}}/M^2$ and more larger samples if not, but the total number of samples should be less than $R^{\mathcal{M}}$.

Remark 5. We can easily convert the RPA attack into the attack on a right-to-left computation using Proposition 5.

5 Comparison

As described in the previous sections, the basic BSD type countermeasures are vulnerable to various attacks. So, the BSD type countermeasures should be combined with additional countermeasures to resist SPA and DPA. In the section, when some countermeasures are added to BSD type countermeasures we compare the proposed attack with previously known attacks introduced by Fouque et al. and Sim et al. and the hidden Markov model (HMM) attack [5] [14] for the left-to-right computation and the right-to-left computation, respectively.

Table 2 shows the possibilities of attacks against several combined countermeasures and the direction of computation. We first consider BSD type countermeasures combined with the SPA countermeasure. Fouque et al. and Sim et al. analyzed the full version of Ha-Moon's method. So, the possibility of their attacks against the other BSD type countermeasures may be determined according to the given random recoding method. However, the attack of Fouque et al. is based on detection of internal data collisions, so their attack may be able to apply without regard to the direction of computation algorithms at a glance. On the other hand, the HMM attack is available under the assumption of distinguishability between ECADD and ECDBL. Thus, the HMM attack seems unable to break the BSD type combined with SPA countermeasures.

[5] The attack introduced by Karlof and Wagner utilizes the hidden Markov model (HMM) to break BSD type countermeasure, which is a cryptanalytic framework for countermeasures that utilizes a probabilistic finite state machine.

Table 2. Comparison of possibility for several combined BSD type countermeasures

Attacks	Left-to-Right computation		Right-to-Left computation		Attack Model
	with SPA C.	with DPA C.	with SPA C.	with DPA C.	
Fouque et al.	Dependent	Infeasible	Dependent	Infeasible	MESD
Sim et al.	Dependent	Infeasible	Infeasible	Infeasible	ZEMD
HMM	Infeasible	Infeasible	Infeasible	Infeasible	HMM
Ours	Feasible	Feasible	Feasible	Feasible	ZEMD

Note. SPA C. and DPA C. denote SPA countermeasure and DPA countermeasure, respectively.
Note. We consider the fixed procedure type such as Coron's dummy method or Montgomery ladder method as an SPA countermeasure, and the randomized point representation type such as randomized projective coordinates or random isomorphisms as a DPA countermeasure.
Note. "Feasible" means that the attack can break all combined countermeasures, "Infeasible" means that the attack can not break any combined countermeasure, and "Dependent" means that the possibility of the attack depends on random recoding methods.

In the case of BSD type countermeasures combined with the DPA countermeasure, the attack of Fouque et al., the attack of Sim et al., and the HMM attack can not break the combined BSD type countermeasures. However, we have shown that BSD type countermeasures combined with DPA countermeasures using randomized point representations such as randomized projective coordinates [4] or random isomorphisms [11] are vulnerable to the proposed RPA attack.

Remark 6. In order to strengthen the security of BSD type countermeasures there are other possible approaches, that is, if BSD type countermeasures are combined with the indistinguishable operations type using the same addition formulae [2] or random point blinding type such as Coron's second method and random initial point method [18], then BSD type countermeasures may be secure against not only the proposed attacks but also the other attacks.

In addition, another difference between the proposed attack and the attack of Fouque et al. is the analysis model. The model of Fouque et al. is based on the collision detection, which is rather different from the usual SCA model. To find collisions their attack utilizes "Multiple Exponent Single Data" (MESD) technique. The MESD requires that an attacker has two identical devices with the same algorithm: one with an unknown secret scalar and the other with a chosen scalar by oneself. In order to recover the unknown secret scalar, we compare the power consumptions of two devices. If the power consumptions are similar then the scalars equal each other, otherwise, the scalars differ. However, such a situation may be less practical than ZEMD technique, which only requires a device with an unknown secret scalar.

6 Concluding Remarks

In this paper, we have enhanced the existing attacks against the full version of Ha-Moon's method, and then we have proposed a practical attack applicable to all BSD type countermeasures combined with an SPA countermeasure. We

showed that the proposed attack is as practical as the original ZEMD attack by the simulations on the target countermeasures.

We have further enhanced the proposed attack in two ways. The proposed attack is extended to a right-to-left computation and BSD type combined with a DPA countermeasure. That is, in order to repair the security, if BSD type countermeasures are combined with a right-to-left computation or DPA countermeasures such as randomized projective coordinates or random isomorphisms, then it may be vulnerable to the proposed attack.

Acknowledgments

Tae Hyun Kim and Jongin Lim were 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 Assessment). Dong-Guk Han was supported by the Korea Research Foundation Grant. (KRF-2005-214-C00016)

References

1. Agagliate, S., Guillot, P., Orcière, O., *A Randomized Efficient Algorithm for DPA Secure Implementation of elliptic curve Cryptosystems*, in the proceedings of Workshop on Coding and Cryptography 2003 (WCC 2003), (2003), 11-19.
2. Brier, É., Joye, M., *Weierstrass Elliptic Curves and Side-Channel Attacks*, Public Key Cryptography (PKC2002), LNCS2274, (2002), 335-345.
3. Chevallier-Mames, B., Ciet, M., Joye, M., *Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity*, IEEE Trans. Computers, Vol.53, No.6, (2004), 760-768.
4. Coron, J.S., *Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems*, Cryptographic Hardware and Embedded Systems (CHES'99), LNCS1717, (1999), 292-302.
5. Ebeid, N., Hasan, A., *Analysis of DPA Countermeasures Based on Randomizing the Binary Algorithm*, Technical Report of the University of Waterloo, No. CORR 2003-14. `http://www.cacr.math.uwaterloo.ca/techreports/2003/corr2003-14.ps`
6. Fouque, P.A., Muller, F., Poupard, G., and Valette, F., *Defeating Countermeasures Based on Randomized BSD Representations*, Cryptographic Hardware and Embedded Systems 2004 (CHES 2004), LNCS3156, (2004), 312-327.
7. Goubin, L., *A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems*, Public Key Cryptography, (PKC 2003), LNCS2567, (2003), 199-211.
8. Ha, J., and Moon, S., *Randomized Signed-Scalar Multiplication of ECC to Resist Power Attacks*, Cryptographic Hardware and Embedded Systems 2002 (CHES 2002), LNCS2523, (2002), 551-563.
9. Han, D.-G., Okeya, K., Kim, T.H., Hwang, Y.S., Park, Y.-H., Jung, S., *Cryptanalysis of the Countermeasures Using Randomized Binary Signed Digits*, Applied Cryptography and Network Security (ACNS'04), LNCS3089, (2004), 398-413.
10. Joye, M., Paillier, P., Schoenmakers, B., *On Second-Order Differential Power Analysis*, Cryptographic Hardware and Embedded Systems (CHES'05), LNCS3659 , (2005), 293-308.

11. 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.
12. Kocher, C., Jaffe, J., Jun, B., *Differential Power Analysis*, Advances in Cryptology - CRYPTO '99, LNCS1666, (1999), 388-397.
13. Koblitz, N., *Elliptic curve cryptosystems*, Math. Comp. 48, (1987), 203-209.
14. Karlof, C., Wagner, D., *Hidden Markov Model Cryptanalysis*, Cryptographic Hardware and Embedded Systems (CHES 2003), LNCS2779, (2003), 17-34.
15. Messerges, T.S., Dabbish, E.A., Sloan, R.H., *Power Analysis Attacks of Modular Exponentiation in Smartcards*, Cryptographic Hardware and Embedded System (CHES 1999), LNCS1717, (1999), 144-157.
16. Messerges, T.S., Dabbish, E.A., Sloan, R.H., *Examining Smart-Card Security under the Threat of Power Analysis Attacks*, IEEE Trans. Computers, Vol.51, No.5, (2002), 541-552.
17. Miller, V.S., *Use of elliptic curves in cryptography*, Advances in Cryptology - CRYPTO '85, LNCS218, (1986), 417-426.
18. Mamiya, H., Miyaji, A., and Morimoto, H., *Efficient Countermeasures Against RPA, DPA, and SPA*, Hardware and Embedded System (CHES 2004), LNCS3156, (2004), 343-356.
19. Montgomery, P. L., *Speeding the Pollard and elliptic curve methods of factorization*, Mathematics of Computation, Vol.48, No.177, (1987), 243-264.
20. Oswald, E., Aigner, M., *Randomized Addition-Subtraction Chains as a Countermeasure against Power Attacks*, Cryptographic Hardware and Embedded Systems (CHES 2001), LNCS2162, (2001), 39-50.
21. Okeya, K., Han, D.-G., *Side Channel Attack on Ha-Moon's Countermeasure of Randomized Signed Scalar Multiplication*, INDOCRYPT 2003, LNCS2904, (2003), 334-348.
22. Okeya, K., Sakurai, K., *Power Analysis Breaks Elliptic Curve Cryptosystems even Secure against the Timing Attack*, INDOCRYPT 2000, LNCS1977, (2000), 178-190.
23. 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.
24. Sim, S.G., Park, D.J., Lee, P.J., *New power analyses on the Ha-Moon algorithm and the MIST algorithm*, Sixth International Conference on Information and Communication Security (ICICS 2004), LNCS3269, (2004), 291-304.
25. Walter, C.D., *Issues of Security with the Oswald-Aigner Exponentiation Algorithm*, The Cryptographers' Track at the RSA Conference 2004 (CT-RSA'04), LNCS2964, (2004), 208-221.
26. Walter, C.D., *Simple Power Analysis of Unified Code for ECC Double and Add*, Cryptographic Hardware and Embedded Systems (CHES'04), LNCS3156 , (2004), 191-204.

A Several Proofs

Proposition 2 *Assume that the computational environment is the same as Proposition 1. If the appearance probability of a discernment point is p, then*

the signal-to-noise ratio is

$$SNR = \frac{\varepsilon p\sqrt{R}}{\sqrt{8\sigma^2 + \varepsilon^2(\alpha M + M - p)}}. \tag{4}$$

Thus the SNR is approximately p times larger than the original. In other words, in order to obtain the same SNR, the required samples are p^{-2} times larger than the original.

Proof. Recall that SNR is defined as the ratio of the average signal divided by its standard deviation.

Since the discernment point appears with the probability p, the signal is p times larger than the original; $E[signal] = \varepsilon p$. If the discernment point does not appear, we can consider such a case as noise; $E[noise] = 0$.

On the one hand, non-algorithmic noise does not depend on the input data. On the other hand, algorithmic noise can be seen as $(M - signal/\varepsilon)$ random bits; $(M - p)$ random bits. Thus, it is easy to see that the variance of the signal is $V[signal] = 4(\sigma^2 + \varepsilon^2(M-p)/4)/R$. If the discernment point does not appear, the variance of the noise is $v[noise] = 4(\sigma^2 + \varepsilon^2\alpha M/4)/R$.

Hence, in the current case, the average is $p\varepsilon$ and the standard deviation is

$$\sqrt{8\sigma^2 + \varepsilon^2(\alpha M + M - p)}/\sqrt{R}.$$

In other words, SNR satisfies the equation (2). □

Proposition 3 *$d_{[i,n]}$ is either $k_{[i,n-1]}$ or $k_{[i,n-1]} + 1$.*

Proof. $d = d_{[i,n]} \cdot 2^i + d_{[0,i-1]}$ and $k = k_{[i,n-1]} \cdot 2^i + k_{[0,i-1]}$. As $d = k$, $(d_{[i,n]} - k_{[i,n-1]}) \cdot 2^i = k_{[0,i-1]} - d_{[0,i-1]}$. As $-2^i < k_{[0,i-1]} - d_{[0,i-1]} < 2^{i+1}$, $-1 < (k_{[0,i-1]} - d_{[0,i-1]})/2^i < 2$. Here, $(k_{[0,i-1]} - d_{[0,i-1]})/2^i$ must be an integer since it is equal to $d_{[i,n]} - k_{[i,n-1]}$. Hence, $d_{[i,n]}$ is either $k_{[i,n-1]}$ or $k_{[i,n-1]} + 1$. □

Proposition 4 *Assume that an attacker can recognize whether the peak occurs or not using $R^{\mathcal{S}}$ samples in the case of* Double_Add_Always method*. If the appearance probability of the discernment point for the target bit k_i is p, the use of $p^{-2}R^{\mathcal{S}}$ samples enables the attacker to recover k_i; $k_i = 0$ if an appreciable peak occurs, or $k_i = 1$ if not.*

Proof. First we discuss the case of $k_i = 0$. When Addition-Subtraction_Always _method manipulates the i-th bit $d_i^{(r)}$, $d_{[i,n]}^{(r)}$ is computed, which is equal to $2 \cdot k_{[i+1,n-1]}$ or $2 \cdot k_{[i+1,n-1]} + 1$ because of Proposition 3, The next iteration of the flow computes $\big(4 \cdot k_{[i+1,n-1]}\big) \cdot P$ or $\big(4 \cdot k_{[i+1,n-1]} + 2\big) \cdot P$. Note that the former is the discernment point. From the assumption, the appearance probability of the discernment point is p. Proposition 2 shows that the use of $p^{-2}R^{\mathcal{S}}$ samples enables the attacker to recognize the peak because of the assumption for his/her capability. Since the peak shows that the attacker's guess is correct, he/she reveals $k_i = 0$. The discussion on the case for $k_i = 1$ is similar. □