# An Improved and Efficient Countermeasure against Power Analysis Attacks

ChangKyun Kim[1], JaeCheol Ha[2], SangJae Moon[3], Sung-Ming Yen[4], Wei-Chih Lien[4], and Sung-Hyun Kim[5]

[1] National Security Research Institute, KOREA
kimck@etri.re.kr
[2] Division of Information Science, Korea Nazarene Univ., KOREA
jcha@kornu.ac.kr
[3] School of Electrical Eng. and Computer Science, Kyungpook National Univ., KOREA
sjmoon@knu.ac.kr
[4] Dept of Computer Science and Information Eng. National Central Univ., TAIWAN.
{yensm,lienwc}@csie.ncu.edu.tw
[5] System LSI Division, Samsung Electronics Co., Ltd., KOREA
teri_kim@samsung.com

**Abstract.** Recently new types of differential power analysis attacks (DPA) against elliptic curve cryptosystems (ECC) and RSA systems have been introduced. Most existing countermeasures against classical DPA attacks are vulnerable to these new DPA attacks which include refined power analysis attacks (RPA), zero-value point attacks (ZPA), and doubling attacks. The new attacks are different from classical DPA in that RPA uses a special point with a zero-value coordinate, while ZPA uses auxiliary registers to locate a zero value. So, Mamiya et al proposed a new countermeasure against RPA, ZPA, classical DPA and SPA attacks using a basic random initial point. His countermeasure works well when applied to ECC, but it has some disadvantages when applied to general exponentiation algorithms (such as RSA and ElGamal) due to an inverse computation. This paper presents an efficient and improved countermeasure against the above new DPA attacks by using a random blinding concept on the message different from Mamiya's countermeasure and show that our proposed countermeasure is secure against SPA based Yen's power analysis which can break Coron's simple SPA countermeasure as well as Mamiya's one. The computational cost of the proposed scheme is very low when compared to the previous methods which rely on Coron's simple SPA countermeasure. Moreover this scheme is a generalized countermeasure which can be applied to ECC as well as RSA system.

**Keywords:** Side channel attack, DPA, RPA, ZPA, doubling attack, SPA, ECC, RSA.

## 1 Introduction

Since P. Kocher introduced power analysis attacks against cryptographic devices [10], many countermeasures have been proposed to prevent power analysis attacks using various hardware and software techniques. Specifically, for elliptic curve cryptosystems (ECC), there are several types of countermeasures including random exponentiation algorithms [2], blinding methods on a point [2, 5], random projective coordinates algorithms [2], and some approaches using special forms of certain elliptic curves (Montgomery form [16], Jacobian form [11], and Hessian form [6]). However, the above countermeasures have some disadvantages; they have a high computational load and some security weaknesses.

First of all, although Coron's countermeasures in [2] seem to provide security against DPA attacks, some papers have shown that these cryptosystems can be broken by new DPA attacks [3]. Also, Coron's first countermeasure increases the computational load as it requires an additional random number $k$. The second countermeasure, the point blinding method, which adds a secret random point R, can also be broken by a doubling attack as proposed by Fouque [3]. Furthermore, most of the widely accepted randomization techniques (Coron's third countermeasure, random elliptic curve isomorphisms and random field isomorphisms) which were thought to protect against differential power analysis attacks (DPA) can be broken by a refined power analysis (RPA) attack as proposed by Goubin [4]. Moreover, an extension of the RPA attack is proposed by T. Akishita *et al* (called ZPA) [1]. None of Coron's three countermeasures with a SPA countermeasure protect against this attack which uses auxiliary registers to find a zero-value.

More recently, two papers have proposed countermeasures to protect against these new DPA attacks. First, Smart analyzed the RPA attack in detail and discounted its effectiveness in a large number of cases [17]. He also presented two defense methods (randomization of the private exponent and point blinding). However, these methods are not efficient from the viewpoint of computational load. Second, Mamiya *et al* proposed a countermeasure (called BRIP) which uses a random initial point (RIP) $R$ [13]. This method, however, is vulnerable to power analysis by exploiting specially chosen input messages [18]. Moreover, it is not a suitable method to be applied to RSA because it requires an inversion computation. To solve the above problems of computational load and vulnerability, this paper presents an improved and efficient countermeasure.

The rest of this paper is organized as follows. In the next section, we summarize the basic operation of ECC and review power analysis attacks including SPA, classical DPA, doubling attacks, RPA, and ZPA. Section 3 presents our proposed countermeasure which uses a random point blinding technique. Section 4 presents an enhancement to protect against SPA, while Section 5 presents an analysis of its performance. Finally, we conclude with Section 6.

## 2 Preliminary

### 2.1 Elliptic Curve Cryptosystems

In 1985 Miller and Koblitz first introduced an elliptic curve cryptosystems, independently [14, 9]. They provided a methology for obtaining high-speed, efficient, and scalable implementations for a secure cryptographic device. The security of ECC depends on the intractability of the elliptic curve analogue of the discrete logarithm problem. This problem has been extensively studied and is well known to be computationally hard.

An elliptic curve is a set of points $(x, y)$ which are solutions of a bivariate cubic equation over a field $K$. An equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where $a_i \in K$, defines an elliptic curve over $K$. As an example, an elliptic curve $E$ defined over the binary field $F_{2^q}$ is transformed to

$$y^2 + xy = x^3 + ax^2 + b \tag{2}$$

with $a, b \in K$. This curve has one point $\mathcal{O}$ at infinity, which is the identity element of the group.

Let $P = (x_1, y_1) \neq \mathcal{O}$ be a point, the inverse of $P$ is $-P = (x_1, -y_1)$. Let $Q = (x_2, y_2) \neq \mathcal{O}$ be a second point with $Q \neq -P$, the doubling of point $P$ is $2(x_1, y_1) = (x_3, y_3)$, where

$$\begin{aligned} x_3 &= \lambda^2 + \lambda + a \\ y_3 &= (x_1 + x_3)\lambda + x_3 + y_1 \end{aligned} \tag{3}$$

with $\lambda = x_1 + (y_1/x_1)$. The addition of two points $P$ and $Q$ is $(x_3, y_3)$, where

$$\begin{aligned} x_3 &= a + \lambda^2 + \lambda + x_1 + x_2 \\ y_3 &= (x_1 + x_3)\lambda + x_3 + y_1 \end{aligned} \tag{4}$$

with $\lambda = (y_1 + y_2)/(x_1 + x_2)$. To subtract the point $P = (x, y)$, one adds the point $-P$.

## 2.2 Power Analysis Attacks

Power analysis attacks are usually divided into two types. The first type, a SPA attack, is based on a single observed power consumption, while the second type, a DPA attack combines a SPA attack with an error-correcting technique using statistical analysis [10]. Most importantly, classical DPA attacks have been extensively researched for each cryptosystem and new types of DPA have been introduced. Many existing countermeasures are vulnerable to the new attacks which include RPA, ZPA, and a doubling attack [3]. In the next section, the above mentioned attacks are described in more detail.

**Simple Power Analysis** A SPA attack consists of observing the power consumption during a single execution of a cryptographic algorithm. The power consumption analysis may also enable one to distinguish between point addition and point doubling in the non-immune Left to Right (L-R) binary method.

To protect against SPA, Coron [2] proposed a simple SPA countermeasure which consisted of modifying the binary method as in Fig. 1. Since none of the instructions in this cryptographic algorithm depend on the data, this algorithm is resistant to a SPA attack. Note that step 2 of the algorithm in Fig. 1 computes a point addition and point doubling without regard to the secret key $d$. However, even though this scheme is resistant to a SPA attack, it remains vulnerable to a DPA attack.

```
      Input: $d$, $P$
      Output: $dP$
1.    $Q[0] = \mathcal{O}$
2.    For $i = n - 1$ downto 0 do:
2.1       $Q[0] = 2Q[0]$.
2.2       $Q[1] = Q[0] + P$.
2.3       $Q[0] = Q[d_i]$.
3.    Return $Q[0]$.
```

**Fig. 1.** Scalar multiplication to resist a SPA attack.

**Classical Differential Power Analysis** A DPA attack is based on the same basic concept as a SPA attack, but uses error correction techniques and statistical analysis to extract very small differences in the power consumption signals. To be resistant to a DPA attack, some system parameters or computation procedures must be randomized. Coron suggested three countermeasures to protect against a classical DPA: randomizing the private exponent, blinding the point $P$, and randomizing the projective coordinates. For Coron's 3rd suggestion, the method of randomizing the projective coordinates, let $P = (X, Y, Z)$ be an elliptic point. Then point $P$ is equal to $(rX, rY, rZ)$ for all $r \in K$, where $r$ is a random number. An enhanced version of Coron's 3rd countermeasure has been proposed by Joye-Tymen [7]. It uses an isomorphism of an elliptic curve, thereby transposing the computation into another curve through a random morphism. The elliptic point $P = (X, Y, Z)$ and parameters ($a$, $b$) of the defined curve equation can be randomized like $(r^2X, r^3Y, Z)$ and ($r^4a$, $r^6b$) for all $r \in K$. However, all of the above countermeasures add computational overhead and are still vulnerable to RPA, ZPA, and doubling attacks as described in the subsections below.

**Doubling Attack** The doubling attack is able to obtain the secret scalar using binary elliptic scalar multiplication [3]. It only works for the L-R binary method. The main idea of this attack is based on the fact that, even if an adversary cannot see whether the computation being done is doubling or addition, he can still detect when the same operation is done twice. More precisely, if a smartcard computes 2A and 2B in any operations, the attacker is not able to guess the value of A or B but he can check if $A = B$ or $A \neq B$. This assumption is reasonable since this kind of computation usually takes many clock cycles and depends greatly on the value of the operands. If the noise is negligible, a simple comparison of the two power traces during the doubling will be efficient to detect this equality.

Two of Coron's three proposed countermeasures against DPA attacks fail to protect against a doubling attack: randomizing the private scalar (exponent) and blinding the point. However, his third countermeasure, the randomized projective coordinate does protect against a doubling attack as does a randomized exponentiation algorithm such as the Ha-Moon algorithm which maps a given scalar to one of various representations. Since the positions of the zeros in the Ha-Moon algorithm vary in each representation, the doubling attack cannot detect the positions of the zeros for the doubling operation.

*Remark 1.* Basically, to protect against a doubling attack, the random blinding point R should be randomly updated. A regularly updated method shouldn't be chosen. A method similar to Coron's 3rd countermeasure or a random field isomorphism should be used.

**Refined Power Analysis Attack** Goubin proposed a new power analysis in 2003, namely the refined power analysis (RPA), which works even if one of the three countermeasures with a SPA countermeasure is applied [4]. The RPA attack assumes that the attacker can input adaptively chosen messages or elliptic curve points to the victim exponentiation algorithm. Smart analyzed the RPA attack in detail and discounted its effectiveness in a large number of order. For the remaining cases Smart proposed a defense against the RPA attack based on isogenies of small degree [17]. However, the RPA attack is still a threat to most elliptic curve cryptosystems.

**Zero-value Point Attack** The zero-value point attack is an extension of the RPA attack [1]. In a RPA attack, the attacker uses a special point which has a zero-value coordinate. In a ZPA attack, on the other hand, he utilizes an auxiliary register which might take a zero-value in the definition field. As a result, Coron's 3rd or random field isomorphism countermeasures do not protect against ZPA attacks.

*Remark 2.* To protect against RPA and ZPA attacks, the base point $P$ or the secret scalar $d$ should be randomized. For example, Coron's first two countermeasures (but not the 3rd) protect against these attacks.

### 2.3 Mamiya's Countermeasure

To protect against these new DPA attacks, Mamiya et al recently proposed a countermeasure (called BRIP) which uses a random initial point (RIP) $R$. He computes $dP + R$ using the simple algorithm depicted in Fig. 2, and subtracts $R$ to get $dP$. In order to protect against SPA, they compute $dP + (1\overline{11}\cdots\overline{11})P$.

```
Input: $d$, $P$
Output: $dP$
$R$ =randompoint()
$T[0] = R,\ T[1] = -R,\ T[2] = P - R$
For $i$ from $n - 1$ downto 0 do
    $T[0] = 2T[0]$
    $T[0] = T[0] + T[d_i + 1]$
Return($Q = T[0] + T[1]$)
```

**Fig. 2.** The binary expansion with RIP(BRIP).

Since BRIP assures that all variables $T[0]$, $T[1]$, and $T[2]$ differ at each execution, no special point or zero-value register will appear during all operations. In the same way, C. K. Kim *et al* proposed a RSA version to protect against power attacks in RSA system [8]. But in their countermeasure (RSA version algorithm) they have to compute an inversion of a random number $r$ with respect to a negative random point $-R$ for ECC. Furthermore, even though we can develop a more computationally efficient countermeasure in which an attacker can not distinguish between a point addition and a point doubling (This assumption comes from [12] and their idea is called by side-channel atomicity). Unfortunately, BRIP is of no use in the side-channel atomicity because it is based on an add-and-double always algorithm.

## 3 New Countermeasure against Side-channel Attacks

In this section, we describe our new countermeasure which is able to protect against existing power attacks including the classic DPA, the RPA, the ZPA, and the doubling attack. Not only does our countermeasure protect against protect against the above attacks, it is also computationally more efficient than existing alternative countermeasures. Finally, our method can be applied to RSA without any problems as it does not require any inversion computations of some integers to be used.

### 3.1 The Proposed Countermeasure

The basic idea of the proposed countermeasure is to blind a point using a random point $R$. We finally compute $dP + \#\varepsilon R$, where $\#\varepsilon$ is the number of points of the curves. Now, let $s = \#\varepsilon - d$, then we compute $d(P + R)$ the related secret scalar $d$ and $sR$ the related an integer $s$. The core of the algorithm is the simultaneous computation of the above two operations $d(P+R)$ and $sR$ as described in Fig. 3. By using a random blinding point technique, the intermediate values of points and registers which are used in each iteration randomly change.

In Fig. 3, to compute $d(P + R)$ and $sR$ simultaneously we modified a multi-scalar multiplication algorithm used in an elliptic curve cryptosystem for signature verification of ECDSA [15]. In this case the final result $dP$ is obtained by computing

$$\begin{aligned} dP &= \sum_i \{d_i(P+R) + s_i R\} \\ &= \sum_i d_i P + \sum_i (d_i + s_i) R \\ &= dP + (d+s)R \\ &= dP + \#\varepsilon R \end{aligned} \tag{5}$$

where $\#\varepsilon R$ is equal to a point $\mathcal{O}$ at infinity.

```
      Input: $d$, $P$
      Output: $dP$
      Pre-computation
1.    $s = \#\varepsilon - d$
2.    Choose a random elliptic point $R$
3.    $T[00] = \mathcal{O},\ T[01] = R,\ T[10] = P + R,$
      $T[11] = P + 2R$
      Evaluation
4.    $Q = T[00]$
5.    For $i$ from $n - 1$ downto 0 do
5.1       $Q = 2Q$
5.2       $Q = Q + T[d_i s_i]$
6.    Return($Q = dP$)
```

**Fig. 3.** The proposed scalar multiplication for ECC.

```
      Input: $d$, $m$
      Output: $m^d$
      Pre-computation
1.    $s = \phi(N) - d$, where $\phi(N)$ is Euler phi-function.
2.    Choose a random number $r$.
3.    $T[00] = 1,\ T[01] = r,\ T[10] = m \cdot r,$
      $T[11] = m \cdot r^2$
      Evaluation
4.    $C = T[00]$.
5.    For $i$ from $n - 1$ downto 0 do
5.1       $C = C^2 \bmod N$
5.2       $C = C \cdot T[d_i s_i] \bmod N$
6.    Return($C = m^d$)
```

**Fig. 4.** The proposed exponentiation algorithm for RSA.

Although an attacker inputs special points to attempt an attack using RPA and ZPA, he cannot bypass the proposed countermeasure because the point $P$ is blinded by the random point $R$ in Eq. 5. The point $R$ is changed at each execution, otherwise can be applied the randomized projective coordinates technique. Therefore, the proposed countermeasure in Fig. 3 satisfies the conditions in remark 1 and remark 2 and can protect against the new DPA attacks (RPA, ZPA, and doubling attacks) as well as classical DPA attacks.

Moreover, our proposed countermeasure can be applied to RSA as in Fig. 4. Notice that it is not necessary to compute an inverse of the random number $r$. It is very important to speed up the RSA computation and implementation. From this point of view, our proposed countermeasure is a more efficient and general method than Mamiya's countermeasure.

## 4 The Low Cost Countermeasure to Resist SPA

In order to protect against SPA, instructions performed during a cryptographic algorithm should not depend on the data being processed. In our proposed countermeasure as in Fig. 5, when $d_i$ and $s_i$ are equal to zero simultaneously, there is no power difference for a pair of digits $d_i s_i$ because $T[00]$ is not a point $\mathcal{O}$ but a certain point which is a 2-torsion point.

```
      Input: $d$, $P$
      Output: $dP$
      Pre-computation
1.    $s = \#\varepsilon - d$
2.    Choose a random elliptic point $R$
3.    $T[00] = G \in E[2],\ T[01] = R + G,\ T[10] = P + R + G,$
      $T[11] = P + 2R + G$
      Evaluation
4.    $Q = T[00]$
5.    For $i$ from $n-1$ downto 0 do
5.1       $Q = 2Q$
5.2       $Q = Q + T[d_i s_i]$
6.    $Q = Q + T[00]$
7.    Return($Q = dP$)
```

**Fig. 5.** The enhanced countermeasure to protect against SPA for ECC.

Let $E$ be an elliptic curve defined over a field $K$. $E[2]$ is defined as follows.

$$E[2] = \{G \in E(\overline{K}) | 2G = \mathcal{O}\} \tag{6}$$

where $\overline{K}$ is an algebraic closure of $K$.

In the pre-computation stage as in step 3, all temporary values $T[ij]$ are added to the point $G$. However the point $G$ does not affect the final result $dP$

and vanishes by a doubling operation (step 5.1) at the $(i+1)$-th iteration. For example, the point $Q_{i+1}$ at step 5.1 is obtained by computing

$$\begin{aligned} Q_{i+1} &= 2Q_i = 2(Q_i + G) = 2Q_i + 2G \\ &= 2Q_i \end{aligned} \tag{7}$$

where $Q_i$ is the result of step 5.2. Although the $T[00]$ is only added to the point $G$, there is little problem to obtain the correct result $dP$ by modifying the step 6 (if $d_0 s_0 = 00$ then $Q = Q + T[00]$). However, this means has some weakness that a conditional jump, the step 6, will definitely leak $d_0$. Therefore, all $T[ij]$ in step 3 should be added by $G$ and this makes the step 6 is unconditionally necessary and leaks nothing.

```
     Input: $d$, $m$
     Output: $m^d$
     Pre-computation
1.   $s = \phi(N) - d$, where $\phi(N)$ is Euler phi-function
2.   Choose a random number $r$
3.   $T[00] = N - 1,\ T[01] = r \cdot (N-1),\ T[10] = m \cdot r \cdot (N-1),$
     $T[11] = m \cdot r^2 \cdot (N-1)$
     Evaluation
4.   $C = T[00]$.
5.   for $i$ from $n-1$ downto 0 do
5.1      $C = C^2 \bmod N$
5.2      $C = C \cdot T[d_i s_i] \bmod N$
6.   $C = C \cdot T[00] \bmod N$
7.   Return($C = m^d$)
```

**Fig. 6.** The enhanced countermeasure to protect against SPA for RSA.

RSA is also implemented in a similar way to ECC. In Fig. 6, The $N-1$ substitutes for $T[00]$ to satisfy $T[00]^2 \equiv 1 \bmod N$. Notice that an inverse $r^{-1}$ is not needed for the enhanced countermeasure to protect against SPA for RSA. In Figs. 5 and 6, the computation amount of the dummy operation to resist SPA is $\frac{d}{4}$, because the probability of $d_i s_i = 00$ is $\frac{1}{4}$.

Now, we consider our countermeasure with respect to running time. As one can see, the proposed algorithm above requires approximately $n$ loop iterations to insert the dummy operations. Such a solution, however, is unsatisfactory from a computational perspective because the insertion of these dummy operations which are required to protect against SPA increases the execution time. In 2004, B. Chevallier-Mames *et al* introduced simple techniques to reduce the computational load in ECC or RSA systems [12]. In this technique, we assume that the squaring in RSA system is processed using the multiplication algorithm in Fig. 7. This algorithm reduces the number of iterations to $1.75n$ operations on average.

So we can save 12.5% in computational load compared to previous methods including Mamiya's countermeasure.

```
     Input: $d$, $m$
     Output: $m^d$
     Pre-computation
1.   $s = \phi(N) - d$, where $\phi(N)$ is Euler phi-function
2.   Choose a random number $r$
3.   $T[00] = 1,\ T[01] = r,\ T[10] = m \cdot r,$
     $T[11] = m \cdot r^2$
     Evaluation
4.   $k = 0, C = T[00]$
5.   While $(i \geq 0)$ do {
5.1      $C = C \cdot T[d_i s_i] \bmod N$
5.2      $k = k \oplus (d_i \vee s_i)$
5.3      $i = i - \neg k$ }
6.   Return$(C = m^d)$
```

**Fig. 7.** The side channel atomic squaring and multiplication exponentiation for RSA.

Fig. 8 shows a side channel atomic doubling and addition multiplication procedure for ECC. In this algorithm, our assumption is that the doubling is processed using the same algorithm as the addition as you see in Fig. 8. As before, we can reduce the number of loop iterations to $1.75n$ operations on average.

```
     Input: $d$, $P$
     Output: $dP$
     Pre-computation
1.   $s = \#\varepsilon - d$
2.   Choose a random elliptic point $R$
3.   $T[00] = \mathcal{O},\ T[01] = R,\ T[10] = P + R,$
     $T[11] = P + 2R$
     Evaluation
4.   $k = 0, Q = T[00]$
5.   While $(i \geq 0)$ do {
5.1      $Q = Q + T[d_i s_i]$
5.2      $k = k \oplus (d_i \vee s_i)$
5.3      $i = i - \neg k$ }
6.   Return$(Q = dP)$
```

**Fig. 8.** The side channel atomic doubling and addition multiplication for ECC.

Note that the step 5.1 in Fig. 8 is either doubling or addition according to $d_i$ and $s_i$. However, this operation should only be operated by an algorithm. For example, we know that the operation for doubling and addition of points is very similar according to section 2. As a result, we can choose either doubling or addition since each operation uses the same successive codes (see Fig. 8). Only two extra field additions are needed for doubling compared to addition. These operations are negligible with respect to computational load. From this fact, an efficient algorithm for step 5.1 in Fig. 8 can be derived as described in Fig. 10 of Appendix.

## 5 Security Consideration

### 5.1 SPA on the BRIP

The BRIP is recently considered effective countermeasures against DPA by using randomization onto the input message and into the computational process of the algorithm in order to remove correlation between the secret key and the power consumption. However, Yen *et al* pointed out a vulnerability of it by exploiting specially chosen input message [18]. Yen's attack can apply to not only Coron's simple SPA countermeasure, but also the BRIP.

In the context of RSA system, given the modulus $N$, an attacker can observe the power trace of $(N-1)^2 \equiv 1 \pmod N$. It can be extended as follows.

$$(N-1)^k \equiv \begin{cases} 1 \pmod N & \text{if } k \text{ is even integer} \\ N-1 \pmod N & \text{if } k \text{ is odd integer.} \end{cases} \tag{8}$$

```
     Input: $d,\ m$
     Output: $C = m^d \bmod n$
1.   $r$ = random integer
2.   $T[0] = r,\ T[1] = r^{-1} \bmod N,\ T[2] = m \cdot r^{-1} \bmod N$
3.   For $i$ from $n-1$ downto 0 do
3.1      $T[0] = T[0]^2 \bmod N$
3.2      $T[0] = T[0] \cdot T[d_i + 1] \bmod N$
4.   Return($C = T[0] \cdot T[1] \bmod N$)
```

**Fig. 9.** BRIP countermeasure for RSA.

Given message $m = N-1$, the exponentiation algorithm in Fig. 9 will have $T[0] = (N-1)^{(d_{n-1}...d_i)_2} \cdot r \bmod N$ after the Step 3.2 of iteration $i$. If $T[0] = r$, then $(d_{n-1}...d_i)_2$ is an even integer and $d_i = 0$. Otherwise, $T[0] = (N-1) \cdot r \bmod N$, $(d_{n-1}...d_i)_2$ is an odd integer and $d_i = 1$.

By observing a collected power trace of performing the algorithm in Fig.9, he can try to identify the value of $T[0]$ (it can only be either 1 or $N-1$) after each iteration and to conduct the aforementioned derivation of each $d_i$.

### 5.2 Is the Proposed Countermeasure Secure against Yen's Attack?

Although our proposed countermeasure is similar in a message blinding method using a random number $r$ to the BRIP, our countermeasure is secure against Yen's attack because the basic concept of the proposed one is not $1 \equiv r \cdot (r^{-1})^{(1\overline{11}\cdots\overline{11})_2} \bmod N$ but $1 \equiv r^{\phi(N)} \bmod N$. More clearly, as shown in the following enumeration, there is only one possible pattern of $C$ at the beginning of each iteration in Fig. 6.

$$\begin{aligned} &\text{Step 5.1:} \quad C = r^{(d_{i-1}+s_{i-1})} \bmod N \\ &\text{Step 5.2:} \quad \forall d_i s_i,\ C = (N-1) \cdot r^{(d_i+s_i)} \bmod N \end{aligned}$$

In the case of BRIP, there are only two possible output values of $C$ (either $r$ or $(N-1) \cdot r \bmod N$) depending on the value of $d_i$ no matter what the value of $C$ was at the beginning of this iteration, while in the case of our proposed countermeasure, there are many possible output values of $C$ depending on the value of $r^{(d_i+s_i)}$. Although there is only one possible pattern of $C$, possible output values of $C$ differ from not only each other but also each iteration. Therefore, our proposed countermeasure is resistant against Yen's attack.

## 6 Conclusion

Most existing countermeasures to protect against DPA attack are vulnerable to the new types of DPA attacks such as RPA, ZPA, and doubling attacks. In addition, the more recently proposed countermeasure, BRIP, is also vulnerable to Yen's SPA-based power analysis. In this paper, our proposed countermeasure protects against the new types of DPA as well as Yen's power analysis. Moreover, the computational cost of the proposed scheme is very low when compared to the previous methods which rely on Coron's simple SPA countermeasure. Notice especially that the proposed countermeasure is a more general countermeasure which can be applied to ECC as well as RSA systems.

## References

1. T. Akishita and T. Takagi, "Zero-value point attacks on elliptic curve cryptosystem," In *Information Security Conference – ISC '03*, LNCS 2851, pp. 218–233, Springer-Verlag, 2003.
2. J. Coron, "Resistance against differential power a nalysis for elliptic curve cryptosystems," In *Cryptographic Hardware and Embedded Systems – CHES '99*, LNCS 1717, pp.292-302, Springer-Verlag, 1999.
3. P.-A. Fouque and F. Valette, "The doubling attack– why upwards is better than downwards," In *Cryptographic Hardware and Embedded Systems – CHES '03*, LNCS 2779, pp. 269–280, Springer-Verlag, 2003.
4. L. Goubin, "A refined power-analysis attack on elliptic curve cryptosystems," In *Public Key Cryptography – PKC '03*, LNCS 2567, pp. 199–210, Springer-Verlag, 2003.

5. J. C. Ha and S. J. Moon, “Randomized signed-scalar multiplication of ECC to resist power attacks,” In *Cryptographic Hardware and Embedded Systems – CHES ’02*, LNCS 2523, pp. 551–563, Springer-Verlag, 2002.
6. M. Joye and J. Quisquater, “Hessian elliptic curves and side-channel attacks,” In *Cryptographic Hardware and Embedded Systems – CHES ’01*, LNCS 2162, pp.402-410, Springer-Verlag, 2001.
7. M. Joye and C. Tymen, “Protections against Differential Analysis for Elliptic Curve Cryptography,” In *Cryptographic Hardware and Embedded Systems – CHES ’01*, LNCS 2162, pp.377-390, Springer-Verlag, 2001.
8. C. K. Kim, J. C. Ha, S. H. Kim, S. K. Kim, S. M. Yen, and S. J. Moon, “A secure and practical CRT-based RSA to resist side channel attacks,” In *Computational Science and Its Applications – ICCSA ’04*, LNCS 3043, pp. 150–158, Springer-Verlag, 2004.
9. N. Koblitz, “Elliptic curve cryptosystems,” In *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, Jan. 1987.
10. P. Kocher, J. Jaffe and B.Jun, “Differential power analysis,” In *Mathematics of ComputationAdvances in Cryptology – CRYPTO ’99*, LNCS 1666, pp. 388-397, Springer-Verlag, 1999.
11. P. Liardet and N. Smart, “Preventing SPA/DPA in ECC systems using the Jacobi form,” In *Cryptographic Hardware and Embedded Systems – CHES ’01*, LNCS 2162, pp. 391-401, Springer-Verlag, 2001.
12. B. C. Mames, M. Ciet, and M. Joye, “Low-cost solutions for preventing simple side-channel analysis: side-channel atomicity,” In *IEEE Transactions on Computers*, vol. 53, No. 6, June 2004.
13. H. Mamiya, A. Miyaji, and H. Morimoto, “Efficient countermeasure against RPA, DPA, and SPA,” In *Cryptographic Hardware and Embedded Systems – CHES ’04*, LNCS 3156, pp. 343–356, Springer-Verlag, 2004.
14. V. Miller, “Uses of elliptic curve in cryptography,” In *Advances in Cryptology – CRYPTO ’85*, LNCS 218, pp. 417-426, Springer-Verlag, 1985.
15. National Institute of Standards and Technology, *Digital Signature Standard*, FIPS 186-2, Feb. 2000.
16. K. Okeya and K. Sakurai, “Power analysis breaks ellptic curve cryptosystems even secure agaisnt the timing attack,” In *Advances in Cryptology – INDOCRYPT ’00*, LNCS 1977, pp.178-190, Springer-Verlag, 2000.
17. N. P. Smart, “An analysis Goubin’s refined power analysis attack,” *Proc. of Cryptographic Hardware and Embedded Systems – CHES ’03*, LNCS 2779, pp. 281–290, Springer-Verlag, 2003.
18. S. M. Yen, W. C. Lien, S. J. Moon, and J. C. Ha, “Power Analysis by Exploiting Chosen Message,” Submitted to *IEE Electronics Letters*, 2004.

# Appendix

Input: $P_1 = (C_1, C_2),\ P_2(C_3, C_4)$
Output: $P_1 + P_2,$ or $2P_1$

| Addition : $P_1 + P_2$ | Doubling : $2P_1$ |
|---|---|
| $C_1 = C_1 + C_3 (= x_1 + x_2)$ | $C_6 = C_1 + C_3$ |
| $C_2 = C_2 + C_4 (= y_1 + y_2)$ | $C_6 = C_3 + C_6 (= x_1)$ |
| $C_5 = C_2 / C_1 (= \lambda)$ | $C_5 = C_2 / C_1 (= y_1 / x_1)$ |
| $C_1 = C_1 + C_5$ | $C_5 = C_1 + C_5 (= \lambda)$ |
| $C_6 = C_5^2 (= \lambda^2)$ | $C_1 = C_5^2 (= \lambda^2)$ |
| $C_6 = C_6 + a (= \lambda^2 + a)$ | $C_1 = C_1 + a (= \lambda^2 + a)$ |
| $C_1 = C_1 + C_6 (= x_3)$ | $C_1 = C_1 + C_5 (= x_3)$ |
| $C_2 = C_1 + C_4 (= x_3 + y_2)$ | $C_2 = C_1 + C_2 (= x_3 + y_1)$ |
| $C_6 = C_1 + C_3 (= x_2 + x_3)$ | $C_6 = C_1 + C_6 (= x_1 + x_3)$ |
| $C_5 = C_5 \cdot C_6$ | $C_5 = C_5 \cdot C_6$ |
| $C_2 = C_2 + C_5 (= y_3)$ | $C_2 = C_2 + C_5 (= y_3)$ |

(a) Side channel atomic elliptic curve.

Input: $Q = (C_1, C_2), T[d_i s_i] = (C_3, C_4)$
Output: $Q = Q + T[d_i s_i] = (C_1, C_2)$

| | |
|---|---|
| 5.1.1 | $l = (d_i \vee s_i)$ |
| 5.1.2 | $C_{6-5l} = C_1 + C_3$ |
| 5.1.3 | $C_{6-4l} = C_{3-l} + C_{6-2l}$ |
| 5.1.4 | $C_5 = C_2 / C_1$ |
| 5.1.5 | $C_{5-4l} = C_1 + C_5$ |
| 5.1.6 | $C_{1+5l} = (C_5)^2$ |
| 5.1.7 | $C_{1+5l} = C_{1+5l} + a$ |
| 5.1.8 | $C_1 = C_1 + C_{5+k}$ |
| 5.1.9 | $C_2 = C_1 + C_{2+2k}$ |
| 5.1.10 | $C_6 = C_1 + C_{6-3k}$ |
| 5.1.11 | $C_5 = C_5 \cdot C_6$ |
| 5.1.12 | $C_2 = C_2 + C_5$ |

(b) Doubling or addition algorithm of step 5.1 in Fig. 8.

**Fig. 10.** Side channel double and addition algorithm.