Random Point Blinding Methods for Koblitz Curve Cryptosystem

Yoo-Jin Baek

While the elliptic curve cryptosystem (ECC) is getting more popular in securing numerous systems, implementations without consideration for side-channel attacks are susceptible to critical information leakage. This paper proposes new power attack countermeasures for ECC over Koblitz curves. Based on some special properties of Koblitz curves, the proposed methods randomize the involved elliptic curve points in a highly regular manner so the resulting scalar multiplication algorithms can defeat the simple power analysis attack and the differential power analysis attack simultaneously. Compared with the previous countermeasures, the new methods are also noticeable in terms of computational cost.

Keywords: Elliptic curve cryptosystem, power attack, countermeasure, Koblitz curve, point blinding.

I. Introduction

Due to its shorter key size and efficient realizations, the elliptic curve cryptosystem (ECC) [1], [2] is getting more popular in securing numerous practical systems. Particularly, while some applications with low power and/or low size such as radio frequency identification (RFID) strongly require some cryptographic mechanisms, the commonly-deployed public key cryptosystems such as RSA may not be adequate for such applications due to the tiny area available for cryptography, and ECC is more promising to such applications [3]. Moreover, since very efficient scalar multiplication algorithms are well known [4], [5], ECC over Koblitz curves is expected to be suitable for applications with low area but high security requirements. However, ECC implementations without consideration for the side-channel attacks are known to be highly susceptible to critical information leakage.

For $a \in \{0,1\}$, the Koblitz curve E_a, the main concern of this paper is the elliptic curve which is defined by the equation $y^2 + xy = x^3 + ax^2 + 1$ over a binary field. The cryptographic use of Koblitz curves was first introduced in [6] and is known to offer significant advantage in the processing time over ordinary elliptic curves. In fact, the National Institute of Standards and Technology (NIST) included Koblitz curves in its recommended curves [7]. This paper proposes efficient new power attack countermeasures for ECC over such Koblitz curves.

Conceptually, an attacker might retrieve some secret information from the power consumption profiling of cryptographic devices [8], unless the devices are designed for addressing adequate countermeasures. Some of these remarkable techniques include the simple power analysis attack (SPA) and the differential power analysis attack (DPA)

Manuscript received June 29, 2009; revised Jan. 3, 2010; accepted Jan. 18, 2010.

Yoo-Jin Baek (phone: +82 31 209 3196, email: yjbaek2@hanmail.net) is with the Department of Smart Card Design, Samsung Electronics, Yongin, Gyeonggi, Rep. of Korea.

doi:10.4218/etrij.10.0109.0378

 © 2010 ETRI

[9], which are suitably applicable to ECC realizations. Fortunately, many researchers proposed elegant countermeasures against these power attacks [9], [10]-[14], and we briefly review some of those which are related to our proposals.

First, [12] proposed new random point blinding methods by using the simple observation that $1 = 2^n - (2^{n-1} + \cdots + 2 + 1)$ for any positive integer n. Thus, for $d = \sum_{i=0}^{n-1} d_i 2^i \in \mathrm{Z}$ and two elliptic curve (EC) points P and R, $dP+R$ can be computed as $dP + R = 2^n R + \sum_{i=0}^{n-1} 2^i (d_i P - R)$. While their methods are directly applicable to Koblitz curves as well, this paper uses the specific properties of Koblitz curves to get more efficient power attack countermeasures. Second, [10] proposed several power attack countermeasures specifically applicable for Koblitz curve cryptosystems, which include the SPA countermeasures and the DPA countermeasures. The first SPA countermeasure, which can be considered as a variant for Koblitz curves of the well known double-and-add always method [9], may, however, be vulnerable to the safe-error attack [15], which is mainly due to its dummy operations' usage. The second SPA countermeasure adopted similar properties of Koblitz curves as the new methods. However, it is only for preventing SPA, while newly proposed methods can defeat DPA as well. The first and third DPA countermeasures are to randomize the scalar, so they are not directly comparable with our proposals since the new methods mainly concern randomizing EC points. Finally, the second DPA countermeasure randomizes a point P by $\tau^r(P)$ for the Frobenius map τ and a randomly chosen r. Hence, it can introduce only a very small amount of randomness since τ^m=1 over $E_a(F_{2^m})$.

This paper is organized as follows. In section II, we briefly overview some basic notations and terminologies of elliptic curves and Koblitz curves. We continue with the basics in section III by going through various power attack methods. In section IV, we present new countermeasures. We remark that the new countermeasures make use of the following special properties of the Koblitz curve $E_a(F_{2^m})$ and the Frobenius map τ which is defined by $\tau(x, y) = (x^2, y^2)$:

1. For any integer d, there is an efficient algorithm of finding $d_i \in \{0,1\}$ such that $d = \sum_{i=0}^{m+a-1} d_i \tau^i$ over the main subgroup of $E_a(F_{2^m})$.
2. $\tau^{m-1} + \cdots + \tau + 1 = 0$ over the main subgroup of $E_a(F_{2^m})$.
3. $\tau^l - (\tau - 1)(\tau^{l-1} + \cdots + \tau + 1) = 1$ for any positive integer l.

In the same section, we also give the detailed analysis of the computational cost and the security aspect of the new countermeasures. In particular, it is shown that, compared with the previous ones, new countermeasures are remarkable in terms of computational cost and strong resistance to various power attack methods including SPA and DPA. For example, securely computing dP over $E_a(F_{2^m})$ in the L-R fashion spends at most m Frobenius-map computations (which can be considered to be free if a normal basis is used for F_{2^m}) and m+2 point additions for a binary version and at most $2^w + \lfloor m/w \rfloor + 2$ point additions and $2^w + \lfloor m/w \rfloor + 1$ τ-computations for a window version of width w. On the contrary, the method in [12] takes about m point doublings and $2^w + \lfloor m/w \rfloor$ point additions and the second SPA countermeasure in [10] spends m + 1 point additions, 1 point doubling and m τ-computations without any measure against DPA.

II. Elliptic Curves

For a power q of a prime p, let F_q and $\overline{F}_q$ denote the finite field with q elements and its algebraic closure, respectively. An elliptic curve E over F_q consists of points $(x, y) \in \overline{F}_q \times \overline{F}_q$ which satisfy the following nonsingular Weierstrass equation:

$$y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6, \quad a_i \in F_q,$$

plus the point at infinity O. For any extension field K of F_q in $\overline{F}_q$, $E(K)$ is defined as

$$E(K) = \{(x, y) \in E \mid x, y \in K\} \bigcup \{O\}.$$

It is well known that E forms an Abelian group under a special addition rule [16].

One of the fundamental operations in ECC is the scalar multiplication for points. For a positive integer k and an elliptic curve point P, the scalar multiplication kP is the operation of adding k copies of P and $(-k)P$ is defined to be $k(-P)$.

1. Koblitz Curves

For $a \in \{0,1\}$, the Koblitz curve E_a is the elliptic curve defined as $y^2 + xy = x^3 + ax^2 + 1$ over a binary field. For E_a to be used for public-key cryptosystems, the group $E_a(F_{2^m})$ should be chosen so that its order is a product of a large prime r and a small integer f. In this setting, $E_a(F_{2^m})$ is called a *group of very nearly prime order* if $r > 2$ and $f = 2$ or 4, and the subgroup of order r is called the *main subgroup* of $E_a(F_{2^m})$ [4].

The Frobenius map τ over $E_a(F_{2^m})$ is defined as $\tau(x, y) = (x^2, y^2)$. If a normal basis is used for the field F_{2^m}, the computation of τ just amounts to two 1-bit left-rotations, which can be considered to be free.

One of the most interesting features of Koblitz curve cryptosystems is that scalars can be efficiently represented as a

τ-adic non-adjacent form (NAF). More precisely, for an integer d, there is an efficient algorithm of finding $d_i \in \{-1,0,1\}$ so that $d_i d_{i+1} = 0$ for each i and $d = \sum_{i=0}^{m+a-1} d_i \tau^i$ over the main subgroup of $E_a(F_{2^m})$ [4], [5]. While the τ-adic NAF has the special advantage over binary τ-adic representations in that its Hamming weight density is lower than that of binary representations, the new countermeasures are using binary τ-adic forms since they do not get any benefit from the lower Hamming weight density. Because we could not find algorithms generating such binary τ-adic forms in the literatures, we add a brief discussion on the subject.

First, consider the following algorithm of converting an integer into a binary τ-adic form.

Algorithm 1. Generating a binary τ-adic form
Input: an integer d.
Output: $d_i \in \{0,1\}$ such that $d = \sum d_i \tau^i$ over the main subgroup of $E_a(F_{2^m})$.
1. Set $c_0 + c_1\tau \leftarrow d \pmod{(\tau^m - 1)/(\tau - 1)}$ using Routine 74 in [4].
2. $\mu \leftarrow (-1)^{1-a}$.
3. $i \leftarrow 0$.
4. While $c_0 \neq 0$ or $c_1 \neq 0$, do
   a. $d_i \leftarrow c_0 \mod 2$,
   b. $(c_0, c_1) \leftarrow (c_1 + \mu(c_0 - d_i)/2, -(c_0 - d_i)/2)$.

To justify algorithm 1, we have to prove the following:

1. $d = \sum d_i \tau^i$ over the main subgroup of $E_a(F_{2^m})$.
2. It terminates in finite steps. More precisely, the length of its output sequence $(d_0, d_1, \cdots)$ is at most $m + a$.

To verify the first claim, we just refer to the following lemma [4].

Lemma 1. In $Z[\tau]$, which is satisfying $\tau^2 - (-1)^{1-a}\tau + 2 = 0$, the following identity holds:

$$(d_0 + d_1\tau)\tau = -2d_1 + (d_0 + (-1)^{1-a} d_1)\tau.$$

To prove that algorithm 1 terminates in finite steps, we first assume that steps 4a and 4b are applied to $c_0 + c_1\tau \in Z[\tau]$ to get $(c_0', c_1') = (c_1 + \mu(c_0 - d_i)/2, -(c_0 - d_i)/2)$. Then, it can easily be verified that the norm of $c_0' + c_1'\tau$ is strictly less than that of $c_0 + c_1\tau$, unless $(c_0, c_1) = (-1, 0)$, in which case algorithm 1 produces the identity $-1 = 1 + \tau + \tau^2$ over $E_0(F_{2^m})$ and $-1 = 1 + \tau + \tau^3$ over $E_1(F_{2^m})$. Finally, the fact that the resulting τ-adic representation length of d is at most $m+a$ can similarly be proved as for the τ-adic NAF [4]. This completes the justification of algorithm 1.

For the scalar multiplication algorithm to resist to power attacks, appropriate measures must also be addressed for all its subroutines including the scalar recoding like algorithm 1 [17]. Particularly, algorithm 1 must be secure to SPA since it is most likely to be applied once for the given scalar. However, there may be two SPA vulnerabilities in the algorithm. First, the routine in step 1 may behave irregularly for the inputs. But, this irregularity can be prevented using, for example, the side-channel atomicity method in [11]. Second, algorithm 1 uses an input-related conditional statement in step 4. However, since the length of the output sequence is at most $m + a$ according to the arguments above, step 4 can be re-written as

4' For i=0 to m, do,

which makes the algorithm behave regularly.

The usual left-to-right (L-R) and right-to-left (R-L) scalar multiplication algorithms can easily be translated to the τ-adic forms. That is, it is enough to replace the point doubling map with the Frobenius map.

We close this section by referring the following proposition from [4].

Proposition. Suppose that $E_a(F_{2^m})$ is a group of very nearly prime order $f \cdot r$. Then, for $P \in E_a(F_{2^m})$,

1) P is in the main subgroup of $E_a(F_{2^m})$ if and only if $P = fQ$ for some $Q \in E_a(F_{2^m})$,
2) $\tau^m(P) = P$ for any $P \in E_a(F_{2^m})$,
3) $(\tau^{m-1} + \cdots + \tau + 1)(P) = O$ for any P in the main subgroup of $E_a(F_{2^m})$.

III. Power Attack

The power attack, which was first introduced by Kocher and others [8], attempts to recover some secret information from power consumption curves. Many kinds of power attack methods against ECC have been proposed so far.

The SPA [9] observes one or a few power signals of cryptographic operations, from which it tries to distinguish between various cryptographic primitives. Accordingly, the following regular τ-adic scalar multiplication algorithms for Koblitz curve cryptosystems may be used as its countermeasure.

Algorithm 2. Regular τ-adic binary L-R method for $E_a(F_{2^m})$
Input: $d = \sum_{i=0}^{m} d_i \tau^i$, P in the main subgroup of $E_a(F_{2^m})$.
Output: $dP \in E_a(F_{2^m})$.
1. $T[0] \leftarrow P$, $T[1] \leftarrow 2P$;
2. $S \leftarrow T[d_m]$.
3. For $i = m - 1$ to 0, do
   a. $S \leftarrow \tau(S) + T[d_i]$.
4. Return S–P.

Algorithm 3. Regular τ-adic binary R-L method for $E_a(F_{2^m})$

Input: $d = \sum_{i=0}^{m} d_i \tau^i$, P in the main subgroup of $E_a(F_{2^m})$.

Output: $dP \in E_a(F_{2^m})$.

1. $T[0] \leftarrow P$, $T[1] \leftarrow 2P$.
2. $S \leftarrow T[d_0]$.
3. $T[0] \leftarrow \tau(T[0])$, $T[1] \leftarrow \tau(T[1])$.
4. For i=1 to m–1, do
   a. $S \leftarrow S + T[d_i]$,
   b. $T[0] \leftarrow \tau(T[0])$, $T[1] \leftarrow \tau(T[1])$.
5. Return $S + T[d_m] - T[0]$.

Note that another regular τ-adic L-R scalar multiplication algorithm for Koblitz curves was proposed in [10]. However, since the algorithm is a variant of the well known double-and-add always method [9] in nature, it may be vulnerable to the safe error attack due to its dummy operations' usage [15]. On the contrary, since all the operations in algorithm 2 and 3 are not dummy operations, the vulnerability above does not concern their security.

In algorithm 3, $T[0]$ will have the value $\tau^m(P) = P$ after step 4, which accounts for the subtraction by $T[0]$ in step 5.

Justification of both algorithms comes from the proposition in section II, which gives the identity: for $d = \sum_{i=0}^{m} d_i \tau^i$

$$dP + P = dP + \sum_{i=0}^{m} \tau^i(P) = \sum_{i=0}^{m} (d_i + 1)\tau^i(P), \qquad (1)$$

for P in the main subgroup of $E_a(F_{2^m})$.

Since additions are always performed in step 3a or 4a, both algorithms are resistant to SPA. Their computational cost can be summarized at most m + 1 point additions, 1 point doubling and m τ-computations for algorithm 2; and at most m + 2 point additions, 1 point doubling and 2m τ-computations for algorithm 3. The newly proposed methods also use the similar properties in (1), but the properties are applied to a random point, not to the base point.

Remark. Algorithm 2 and 3 were described in a general setting for a. However, if we only consider the case of a=0, we can get more efficient versions of the algorithms. That is, if a=0, then the τ-adic representation size of d is at most m. Thus, instead of (1), we can use the relation $dP = \sum_{i=0}^{m-1} (d_i + 1)\tau^i(P)$.

The resulting algorithms can then save 1 point addition for algorithm 2 (since we can skip the subtraction in step 4) and 2 point additions and τ-computations for algorithm 3 (since we can skip step 4 for $i = m - 1$ and the subtraction by $T[0]$ in step 5).

The Montgomery method [13] and the side-channel atomicity method [11] can also be used as SPA countermeasures.

The DPA [9] collects a large number of power curves and uses advanced signal-processing techniques to get some useful information from the curves. As countermeasures, the random exponent blinding, the random point blinding, and the random coordinate blinding may be used for ECC [9]. The new countermeasures mainly deal with the random point blinding method to defeat DPA. The higher-order DPA [18], which investigates the statistical relation between several sample points in power curves, may be another big threat to cryptosystem's implementations. However, combining the point blinding and the scalar blinding appropriately may help resist the attack.

The doubling attack [19] is based on the hypothesis that an attacker can distinguish the equality between two intermediate results of two distinct cryptographic operations by analyzing their power curves. Also, it can be defeated if a random point blinding technique is used in an appropriate manner.

The refined power-analysis attack (RPA) [20] chooses a special point such that one of its coordinates is equal to 0 and then inputs into the target device the point which is equal to the special point when it is multiplied by a specific scalar. The zero-value point attack (ZPA) [21] refines RPA and uses the zero-value register, not the zero-value coordinate for the attack. Both RPA and ZPA work if the intermediate results of corresponding scalar multiplications can be correctly guessed. Hence, if an appropriate random point blinding technique is used, these attacks can be prevented.

Finally, applied to the countermeasures in [14], the N-1 attack can work only if the randomization factor doesn't vary during the computation, which is not the case for the new countermeasures.

IV. New Power Attack Countermeasures for Koblitz Curve Cryptosystems

For cryptographic use, the underlying group is preferred to have a prime order, so all the points in consideration are assumed to be contained in the main subgroup of $E_a(F_{2^m})$ hereafter.

Since new countermeasures inevitably require generating random elliptic curve points, we first address this issue. While generating random points over a prime field usually requires an expensive square-root finding algorithm (for example, if we are working in a prime field GF(p), computing $g^{1/2}$, if any, corresponds to computing g^{k+1} if p is of the form $p = 3k + 4$ [22]), it is not so expensive over binary fields. To be concrete, we present an exemplary algorithm of generating a random point on Koblitz curves in the following [22].

Algorithm 4. Random point generation over Koblitz curves

Input: $E_a(F_{2^m})$.

Output: a random point $R(\neq O)$ in the main subgroup of $E_a(F_{2^m})$.

1. Choose a non-zero random element $x \in F_{2^m}$.
2. $\alpha \leftarrow x^3 + ax^2 + 1$.
3. If α=0, go to step 1.
4. $\beta \leftarrow x^{-2}\alpha$.
5. Find z for which $z^2 + z = \beta$, if any. If there is no such z, go to step 1.
6. Set $y = (z+\mu)x$ for a random bit μ.
7. Return R=(x, y).

Note that the original algorithm in [22] outputs a random point in the group $E_a(F_{2^m})$, while algorithm 4 randomly generates a point in the main subgroup of $E_a(F_{2^m})$.

Since the probability of having z for which $z^2 + z = \beta$ is about 1/2 for a random field element β, algorithm 4 takes 4 field multiplications and 1.5 field inversions on average (and some field additions and squarings, which can be considered to be free if a normal basis is used for F_{2^m}). Note that algorithm 4 is highly regular since the conditional statements in step 3 and step 5 are only for excluding some points as an output, and a solution z of the equation in step 5 can be found using a regular routine in [22]. In conclusion, algorithm 4 is highly resistant to SPA.

Finally, the following lemma justifies algorithm 4.

Lemma 2. If m is a prime number (which must be satisfied for the cryptographic use of $E_a(F_{2^m})$ [4]), then the output point (x, y) of algorithm 4 is contained in the main subgroup of $E_a(F_{2^m})$.

Proof. Note that only O, (0,1), (1,0), (0,1) may be contained in the main subgroup of $E_a(F_{2^m})$ [4]. Now, since $x \neq 0$ by step 1, (0, 1) is excluded for the output. On the other hand, α=0 in step 3 only if a=0 and x=1, since m is a prime number. Thus, (1, 0) and (1, 1) cannot be outputted as well. □

Another technique of generating random points on Koblitz curves can be found in [23], which combines the pre-computed random points in a specific way and also discusses the output probability distribution of the resulting random points.

Finally, to blind the computation $Q = dP$ by $Q = d(P+R)–S$, the authors of [9], [24] proposed to store an initial random point pair (R, S) with $S = dR$ and to update the pair for later use by multiplying R and S by a small (random) scalar. However, the (random) scalar for the updating procedure must carefully be chosen to resist to some side-channel attacks [25]. Also, this method as well as the one in [23] requires some memory space for storing the pre-computed values, which is undesirable in some memory-constrained applications.

1. L-R Versions

For the L-R binary versions of new power attack countermeasures, we first observe the identity $\tau^l - (\tau - 1)(\tau^{l-1} + \cdots + \tau + 1) = 1$ for any positive integer l and represent $R \in E_a(F_{2^m})$ as $R = \tau^l(R) - (\tau^{l-1} + \cdots + \tau + 1)(\tau(R) - R)$.

Thus, for $P, R \in E_a(F_{2^m})$ and $d = \sum_{i=0}^{l-1} d_i \tau^i$, we have

$$dP + R = \tau^l(R) + \sum_{i=0}^{l-1} \tau^i (d_i P - \tau(R) + R), \qquad (2)$$

and can obtain the following algorithm.

Algorithm 5.

Input: $d = \sum_{i=0}^{l-1} d_i \tau^i$, $P \in E_a(F_{2^m})$

Output: $dP \in E_a(F_{2^m})$

1. Choose a random point $R(\neq O)$ using algorithm 4.
2. $T[0] \leftarrow R - \tau(R)$, $T[1] \leftarrow T[0] + P$.
3. $S \leftarrow R$.
4. For i=l–1 to 0, do
   a. $S \leftarrow \tau(S) + T[d_i]$.
5. Return S–R.

Algorithm 5 has a very similar structure with the methods in [12]. Actually, the only difference between them lies in the usage of the Frobenius map in algorithm 5 instead of the point doubling map in [12].

Since it always performs additions in step 4a regardless of the scalar bit d_i, algorithm 5 defeats SPA. Also, its resistance to DPA, RPA, and ZPA comes from the randomization of the register S. That is, one cannot guess the intermediate values of S due to the randomization. So DPA, RPA, and ZPA can not be applied to the algorithm.

The computational cost of algorithm 5 is at most l+3 point additions and l+1 τ-computations plus the effort of generating a random point R. Thus, compared with algorithm 2, algorithm 5 can prevent both SPA and DPA, using only 3 additional point additions, one less point doubling, 2 additional τ-computations, and the effort for generating a random point.

One advantage of algorithm 4 is that it can work for any τ-adic length of d. Hence, it can easily adopt scalar blinding methods (to give a better resistance to DPA), even though the scalar blindings cause an increase of the representation length. However, if the representation length of the given scalar is fixed to m+1, then a more efficient point blinding algorithm can be obtained using the identity $\tau^m + \tau^{m-1} + \cdots + 1 = 1$ over the main subgroup of $E_a(F_{2^m})$. That is, we can obtain

$$dP + R = \sum_{i=0}^{m} \tau^i (d_i P + R), \qquad (3)$$

for $d = \sum_{i=0}^{m} d_i \tau^i$ and R in the main subgroup, and the

Table 1. Comparison of algorithms' cost for L-R versions.

	Computational cost
Alg. 2	m+1 point additions, 1 point doubling, m τ-computations
Alg. 5	l+3 point additions, l+1 τ-computations (plus cost for generating a random point)
Alg. 6	m+2 point additions, m τ-computations (plus cost for generating a random point)

corresponding algorithm can be obtained as follows:

Algorithm 6.
Input: $d = \sum_{i=0}^{m} d_i \tau^i$, $P \in E_a(F_{2^m})$.
Output: $dP \in E_a(F_{2^m})$.
1. Choose a random point $R(\neq O)$ using algorithm 4.
2. $T[0] \leftarrow R$, $T[1] \leftarrow P + R$.
3. $S \leftarrow T[d_m]$.
4. For i=m–1 to 0, do
   a. $S \leftarrow \tau(S) + T[d_i]$.
5. Return S–R.

The security aspect of algorithm 6 can similarly be analyzed as in algorithm 5. Its computational cost is at most m+2 point additions and m τ-computations plus the effort of generating R. Thus, compared with algorithm 5 with $l = m + 1$, algorithm 6 can save two point additions and τ-computations. Also, compared with algorithm 2, algorithm 6 can defeat SPA and DPA simultaneously, using only one additional point addition, one less point doubling and the effort for generating R. We emphasize that, as in the remark in section III, a more efficient version of algorithm 6 can be obtained for $a = 0$ if we use the relation $\tau^{m-1} + \cdots + 1 = 0$ over the main subgroup. We summarized these cost-related discussions in Table 1.

Next, for extending algorithm 5 to the window method of width w, we first assume that the binary τ-adic length l of d is divisible by w (by appending some 0 bits in the most significant position, if necessary) and modify (2) as follows: for $l = wt$,

$$\begin{aligned} dP + R &= \sum_{i=0}^{l-1} d_i \tau^i(P) + \tau^l(R) + \sum_{i=0}^{l-1} \tau^i(R - \tau(R)) \\ &= \tau^l(R) + \sum_{u=0}^{t-1} \sum_{j=0}^{w-1} \tau^{wu+j}(d_{wu+j} P + R - \tau(R)) \\ &= \tau^l(R) + \sum_{u=0}^{t-1} \tau^{wu} \left[R - \tau^w(R) + \sum_{j=0}^{w-1} d_{wu+j} \tau^j(P) \right], \end{aligned}$$

for $P, R \in E_a(F_{2^m})$. Also, for extending algorithm 6, we modify (3) into $dP = \sum_{i=0}^{m} d_i \tau^i(P) + \sum_{i=0}^{m-1} \tau^i(R - \tau(R))$ to derive

$$\begin{aligned} dP &= \sum_{i=wt}^{m} d_i \tau^i(P) + \sum_{i=wt}^{m-1} \tau^i(R - \tau(R)) \\ &\quad + \sum_{i=0}^{wt-1} \tau^i(d_i(P) + R - \tau(R)) \\ &= \sum_{i=wt}^{m} d_i \tau^i(P) + R - \tau^k(R) \\ &\quad + \sum_{u=0}^{t-1} \tau^{wu} \left[R - \tau^w(R) + \sum_{j=0}^{w-1} d_{wu+j} \tau^j(P) \right], \end{aligned} \tag{4}$$

for $m+1 = wt + k$, $0 \le k < w$ and $P, R \in E_a(F_{2^m})$. Note that, in (4), we used R–$\tau(R)$ instead of R for a random point since it is easier to calculate $R - \tau^w(R)$ (which appears in the last part of (4)) than $R + \tau(R) + \cdots + \tau^{w-1}(R)$. The validity of using R–$\tau(R)$ instead of R comes from the following lemma.

Lemma 3. If R is randomly chosen from the main subgroup of $E_a(F_{2^m})$, then R–$\tau(R)$ is also randomly distributed in the subgroup.

Proof. If R is in the main subgroup of $E_a(F_{2^m})$, then $R = fQ$ for some $Q \in E_a(F_{2^m})$ by the proposition in section II, so R–$\tau(R)$ = f(Q–τ(Q)), which implies that R–$\tau(R)$ is also contained in the subgroup. To show that R–$\tau(R)$ is randomly distributed, it is sufficient to show that the map $1-\tau$ is a bijection over the main subgroup. In addition, since $1-\tau$ is a group homomorphism, the bijectivity can be proved by showing that R–$\tau(R) = O$ if, and only if, $R = O$. The last claim can easily be verified using the fact that R–$\tau(R) = O$ if, and only if, $R = O$, (0,0), (0,1), (1,0), (1,1). Note that the points (0,0), (0,1), (1,0), (1,1) are not contained in the main subgroup of $E_a(F_{2^m})$. □

Finally, for the extensions to the window method with width w, it is required to generate a table $T[i]$, $i = 0, \cdots, 2^w - 1$ with $T[2^{j_1} + \cdots + 2^{j_k}] = T[0] + \tau^{j_1}(P) + \cdots + \tau^{j_k}(P)$ for $0 < 2^{j_1} + \cdots + 2^{j_k} \le 2^w - 1$.

Algorithm 7. Generating a table T
Input: P, $T[0] \in E_a(F_{2^m})$, w.
Output: $T[j] \in E_a(F_{2^m})$, $j = 1, \cdots, 2^w - 1$ so that
$T[2^{j_1} + \cdots + 2^{j_k}] = T[0] + \tau^{j_1}(P) + \cdots + \tau^{j_k}(P)$ for
$0 < 2^{j_1} + \cdots + 2^{j_k} \le 2^w - 1$.
1. $T[2^j] \leftarrow T[0] + \tau^j(P)$ for $j = 0, \cdots, w-1$.
2. For $i = 2$ to w.
   a. For $0 \le j_1 < \cdots < j_i \le w - 1$
   $T[2^{j_1} + \cdots + 2^{j_i}] = T[2^{j_1} + \cdots + 2^{j_{i-1}}] + \tau^{j_i}(P)$.
3. Return $\{T[j] \mid j = 1, \cdots, 2^w - 1\}$.

Algorithm 7 takes 2^w–1 EC point additions and τ-computations for its execution. Now, we can get the following

algorithms:

Algorithm 8. 2^w-ary version of algorithm 5

Input: $d=\sum_{i=0}^{l-1} d_i\tau^i$, $P\in E_a(F_{2^m})$, w.

Output: $dP\in E_a(F_{2^m})$.

1. Append 0 bits, if necessary, to make the new representation length l' of d be divisible by w. Let $l'=wt$.
2. Choose a random point $R(\neq O)$ using algorithm 4.
3. $S\leftarrow R$.
4. $T[0]\leftarrow R-\tau^w(R)$.
5. Compute $T[i]$, $i=1,\cdots,2^w-1$, using algorithm 7.
6. For $u=t-1$ to 0, do
   a. $S\leftarrow \tau^w(S)+T[\sum_{j=0}^{w-1} d_{wu+j}2^j]$.
7. Return $S-R$.

Algorithm 9. 2^w-ary version of algorithm 6

Input: $d=\sum_{i=0}^{m} d_i\tau^i$ with $m=wt+k$ and $0\le k<w$, $P\in E_a(F_{2^m})$, w.

Output: $dP\in E_a(F_{2^m})$.

1. Choose a random point $R(\neq O)$ using algorithm 4.
2. $T[0]\leftarrow R-\tau^w(R)$.
3. Compute $T[i]$, $i=1,\cdots,2^w-1$ using algorithm 7.
4. $S\leftarrow T[\sum_{j=0}^{k} d_{wt+j}2^j]+\tau^w(R)-\tau^k(R)$.
5. For $u=t-1$ to 0, do
   a. $S\leftarrow \tau^w(S)+T[\sum_{j=0}^{w-1} d_{wu+j}2^j]$.
6. Return S.

The computational cost of both algorithms can be summarized as: assuming that τ^j-computation for $j>1$ can be computed with the same cost as that for τ-computation, algorithm 8 takes at most $2^w+\lceil l/w\rceil+1$ point additions and $2^w+\lceil l/w\rceil$ τ-computations, and algorithm 9 takes at most $2^w+\lfloor m/w\rfloor+2$ point additions and $2^w+\lfloor m/w\rfloor+1$ τ-computations.

2. R-L Versions

Even though R-L scalar multiplication algorithms are very difficult to adopt the window method and usually require more registers than their L-R counterpart, they also possess some merits. For example, to convert an ordinary integer to a τ-adic form, algorithm 1 is working on the right-to-left manner, that is, the least significant bit is first calculated. Hence, the new countermeasures in the R-L fashion may be preferable in the memory-constrained environment since there is no need for wholly storing the newly recoded scalar. This point is particularly critical when the newly recoded scalar cannot be overwritten to the memory space for the original scalar, for example, when the original scalar is in the read-only memory. Also, the authors of [19] commented that the R-L scalar multiplication algorithms may be stronger against some specific power attacks than the L-R algorithms. With these observations in mind, we present the R-L versions of previous algorithms. The resulting algorithms can similarly be shown to be resistant to all the power attacks mentioned before, partially due to the fact that they are based on the R-L scalar multiplication algorithm and partially due to the fact that they make the input point randomly blinded.

Algorithm 10. R-L counterpart of algorithm 5

Input: $d=\sum_{i=0}^{l-1} d_i\tau^i$, $P\in E_a(F_{2^m})$.

Output: $dP\in E_a(F_{2^m})$.

1. Choose a random point $R(\neq O)$ using algorithm 4.
2. $S\leftarrow \tau^l(R)$.
3. $T[0]\leftarrow \tau(R)-R$, $T[1]\leftarrow T[0]+P$.
4. For $i=0$ to $l-2$, do
   a. $S\leftarrow S+T[d_i]$.
   b. $T[0]\leftarrow \tau(T[0])$, $T[1]\leftarrow \tau(T[1])$.
5. Return $S+T[d_{l-1}]-R$.

Algorithm 11. R-L counterpart of algorithm 6

Input: $d=\sum_{i=0}^{m} d_i\tau^i$, $P\in E_a(F_{2^m})$.

Output: $dP\in E_a(F_{2^m})$.

1. Choose a random point $R(\neq O)$ using algorithm 4.
2. $T[0]\leftarrow R$, $T[1]\leftarrow P+R$.
3. $S\leftarrow T[d_0]$.
4. $T[0]\leftarrow \tau(T[0])$, $T[1]\leftarrow \tau(T[1])$.
5. For $i=1$ to $m-1$, do
   a. $S\leftarrow S+T[d_i]$,
   b. $T[0]\leftarrow \tau(T[0])$, $T[1]\leftarrow \tau(T[1])$.
6. Return $S+T[d_m]-T[0]$.

Clearly, algorithm 11 can be simplified further for $a=0$, using the argument as in the remark in section III. The computational cost of algorithm 10 and 11 are summarized along with algorithm 3 in Table 2.

V. Conclusion

In this paper, we proposed new efficient power attack countermeasures for Koblitz curve cryptosystems which are based on the random point blinding technique and the special properties of Koblitz curves. We further favorably extended these techniques to the window method. Our study included

Table 2. Comparison of algorithms' cost for R-L versions.

	Computational cost
Alg. 3	m+1 point additions, 1 point doubling, $2m$ τ-computations
Alg. 10	l+3 point additions, $2l$ τ-computations (plus cost for generating a random point)
Alg. 11	m+2 point additions, $2m$ τ-computations (plus cost for generating a random point)

the detailed investigation into the computational cost and the security aspect for the new countermeasures.

References

[1] N. Koblitz, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, 1987, pp. 203-209.

[2] V.S. Miller, "Use of Elliptic Curves in Cryptography," *CRYPTO, LNCS*, vol. 218, 1986, pp. 417-426.

[3] L. Batina et al., "An Elliptic Curve Processor Suitable for RFID-Tags," *IACR Cryptology ePrint Archive*, 2008.

[4] J. Solinas, "Efficient Arithmetic on Koblitz Curves," *Designs, Codes and Cryptography*, vol. 19, 2000, pp. 195-249.

[5] W. Meier and O. Staffelbach, "Efficient Multiplication on Certain Nonsupersingular Elliptic Curves," *CRYPTO, LNCS*, vol. 740, 1992, pp. 333-344.

[6] N. Koblitz, "CM-Curves with Good Cryptographic Properties," *CRYPTO, LNCS*, vol. 576, 1991, pp. 279-287.

[7] NIST FIPS 186-2, *Recommended Elliptic Curves for Federal Government Use, Appendix to FIPS 186-2*, National Institute of Standards and Technology, 2000.

[8] P. Kocher, J. Jaffe and B. Jun, "Differential Power Analysis," *CRYPTO, LNCS*, vol. 1666, 1999, pp. 388-397.

[9] J. Coron, "Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems," *CHES, LNCS*, vol. 1717, 1999, pp. 292-302.

[10] M. Anwar Hasan, "Power Analysis Attacks and Algorithmic Approaches to their Countermeasures for Koblitz Curve Cryptosystems," *CHES, LNCS*, vol. 1965, 2000, pp. 93-108.

[11] B. Mames, M. Ciet, and M. Joye, "Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity," *IEEE Trans. Computers*, vol. 53, no. 6, 2004, pp. 760-768.

[12] H. Mamiya, A. Miyaji, and H. Morimoto, "Secure Elliptic Curve Exponentiation against RPA, ZPA, DPA, and SPA," *IEICE Trans. Fundamentals*, vol. E89-A, no. 8, 2006, pp. 2207-2215.

[13] P. Montgomery, "Speeding the Pollard and Elliptic Curve Methods for Factorizations," *Mathematics of Computation*, vol. 48, 1987, pp. 243-264.

[14] K. Okeya, T. Takagi, and C. Vuillaume, "Efficient Representations on Koblitz Curves with Resistance to Side Channel Attacks," *ACISP, LNCS*, vol. 3574, 2005, pp. 218-229.

[15] S.M. Yen and M. Joye, "Checking Before Output May Not Be Enough Against Fault-Based Cryptanalysis," *IEEE Trans. Computers*, vol. 49, 2000, pp. 967-970.

[16] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer, 1993.

[17] M. Joye and M. Tunstall, "Exponent Recoding and Regular Exponentiation Algorithms," *AfricaCrypt., LNCS*, vol. 5580, 2009, pp. 334-349.

[18] T. Messerges, "Using Second-Order Power Analysis to Attack DPA Resistant Software," *CHES, LNCS*, vol. 1965, 2000, pp. 238-251.

[19] P. Fouque and F. Valette, "The Doubling Attack - Why Upwards Is Better Than Downwards," *CHES, LNCS*, vol. 2779, 2003, pp. 269-280.

[20] L. Goubin, "A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems," *PKC, LNCS*, vol. 2567, 2003, pp. 199-210.

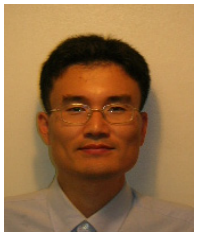
[21] T. Akishita and T. Takagi, "Zero-value Point Attacks on Elliptic Curve Cryptosystem," *ISC, LNCS*, vol. 2851, 2003, pp. 218-233.

[22] IEEE Std. P1363, *IEEE P1363: IEEE Standard Specifications for Public-Key Cryptography*, IEEE, 2000.

[23] J. Coron, D. M'Raihi, and C. Tymen, "Fast Generation of Pairs (k, [k]P) for Koblitz Elliptic Curves," *SAC, LNCS*, vol. 2259, 2001, pp. 151-164.

[24] P.C. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *CRYPTO, LNCS*, vol. 1109, 1996, pp. 104-113.

[25] P. Fouque and F. Valette, "The Doubling Attack - Why Upwards Is Better Than Downwards," *CHES, LNCS*, vol. 2779, 2003, pp. 269-280.



Yoo-Jin Baek received the BE, ME, and PhD degrees in mathematics from Seoul National University, Seoul, Korea, in 1997, 1999, and 2003, respectively. Since July 2003, he has worked for Samsung Electronics. He was involved in projects of securely implementing AES, RSA, and ECC in software and hardware at Samsung Electronics. He has also conducted research in the areas of cryptography and side-channel attacks. He is currently a member of the KIISC and IEEK.