Construction of Efficient and Secure Pairing Algorithm and Its Application

Dooho Choi, Dongguk Han, and Howon Kim

Abstract: **The randomized projective coordinate (RPC) method applied to a pairing computation algorithm is a good solution that provides an efficient countermeasure against side channel attacks. In this study, we investigate measures for increasing the efficiency of the RPC-based countermeasures and construct a method that provides an efficient RPC-based countermeasure against side channel attacks. We then apply our method to the well-known η_T pairing algorithm over binary fields and obtain an RPC-based countermeasure for the η_T pairing; our method is more efficient than the RPC method applied to the original η_T pairing algorithm.**

Index Terms: **Differential power analysis (DPA), Eta pairing, randomized projective coordinate (RPC), side channel attacks (SCAs), Tate pairing.**

I. INTRODUCTION

Pairings on elliptic curves are a well-known subject in the field of cryptography. They have been applied to many cryptographic schemes, such as identity-based encryption [1], [2], identity-based signature [3]–[5], tripartite key agreement [6], short signature [7], and identity-based authentication key agreement [8]. Incidentally, pairings on elliptic curves were first introduced as cryptanalytic tools in [9], [10].

Since the main difficulty in the efficient implementation of pairing-based cryptographic schemes is the computation of then pairing, many techniques have been developed for pairing computation. Barreto *et al.* [11] and Galbraith *et al.* [12] have provided techniques for efficient computations of the pairings by eliminating the unnecessary computation steps from the original Miller's algorithm [13]. Duursma and Lee [14] have found a closed formula of the Tate pairing over a field with a characteristic value of three. Kwon [15] also provided a closed formula over a field with a characteristic value of two. To shorten the main loop of the Tate pairing computation, Barreto *et al.* [16] have defined the Eta pairing on some supersingular curves. Hess *et al.* [17] extended this in a more generic manner to the Ate pairing on non-supersingular elliptic curves.

Side channel attacks (SCAs) commonly utilize a relation between side-channel information related to secret data and internal values during cryptographic operations [18], [19]. There has been some progress in the research on SCAs on pairing-computation algorithms in the works of Page and Vercauteren [20], Whelan and Scott [21], and Kim *et al.* [22]. In [22], Kim *et al.* have investigated the security of the η_T pairing over binary fields in the context of SCAs.

A number of countermeasures have already been developed to protect pairing algorithms against SCAs [20]–[23]. In [20], the bilinearity of pairing is utilized to hide the secret information. Scott [23] has proposed a very simple concept in which the Miller variable m in the BKLS algorithm [11] is multiplied by a random element that will be eliminated in the final exponentiation. In [21], it is remarked that the random value must be multiplied not only by the Miller variable, but also by all intermediate values that comprise the Miller variable in order to obtain an efficient countermeasure. In [22], Kim *et al.* have directly applied the randomized projective coordinate (RPC) method to the original η_T pairing algorithm developed by Barreto *et al.* [16], and have shown that their countermeasure is the most efficient among all the existing countermeasures.

In this paper, for a given extension field equation, we first perform a measurement to estimate the computation cost of an extension field equation to which the RPC method has been applied. Then, we propose a method for constructing an efficient and secure pairing algorithm from a given pairing algorithm. To demonstrate the application of our method, we present an efficient RPC-based countermeasure of the η_T pairing over a binary field, which reduces the additional computation cost by 17% as compared to Kim *et al.*'s countermeasure in [22].

This paper is organized as follows. Section II briefly introduces the definitions of the Tate and η_T pairings. Section III describes the SCAs and its countermeasures on pairing algorithms. Section IV describes a measurement to estimate the efficiencies of RPC-based countermeasures and proposes a method for the development of an efficient countermeasure against a differential power analysis (DPA) attack; in Section V, we apply our construction method to the well-known η_T pairing algorithm over binary fields. The last section presents the conclusions of this study.

II. PAIRINGS ON ELLIPTIC CURVES

Let E be an elliptic curve over a finite field $\mathbb{F}_q$; l be a positive integer coprime to q, which divides $\#E(\mathbb{F}_q)$; and k be the smallest positive integer such that $l|(q^k-1)$ (this is termed the *embedding degree*). The Tate pairing of order l is defined as follows:

$$\tau_l : E(\mathbb{F}_q)[l] \times E(\mathbb{F}_{q^k})[l] \to \mu_l \text{ by}$$
$$(P, Q) \mapsto f_{l,P}(\mathcal{D}_Q)^{(q^k-1)/l}$$

where $f_{l,P}$ is a rational function such that its principal divisor $(f_{l,P})$ is equivalent to $l(P) - (lP) - (l-1)(\mathcal{O})$, $\mathcal{D}_Q$ is a zero

Manuscript received July 31, 2007; approved for publication by Chae Hoon Lim, Division I Editor, December 05, 2007.

This work has been supported by the IT R&D program of MIC/IITA, [2005-S088-03, Development of Security technology for Secure RFID/USN Service].

D. Choi and D. Han are with the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, email: {dhchoi,christa}@etri.re.kr.

H. Kim is a corresponding author of this manuscript, and he is with the Department of Computer Engineering, Pusan National University Busan, Korea, email: howonkim@pusan.ac.kr.

divisor equivalent to $(Q) - (\mathcal{O})$ such that $\mathcal{D}_Q$ has disjoint support with $(f_{l,P})$, and μ_l is the group of the lth roots of unity in $\mathbb{F}_{q^k}^*$. The Tate pairing can be essentially computed by using the following Miller's formula [11], [13];

$$f_{a+b,P}(\mathcal{D}_Q) = f_{a,P}(\mathcal{D}_Q) f_{b,P}(\mathcal{D}_Q) \frac{\ell_{aP,bP}(\mathcal{D}_Q)}{\upsilon_{(a+b)P}(\mathcal{D}_Q)}$$

where $\ell_{aP,bP}$ is a line through points aP and bP (it is a tangent line at aP if $a = b$), and $\upsilon_{(a+b)P}$ is a vertical line at the point $(a+b)P$.

Barreto *et al.* [11] showed that $\tau_l(P,Q) = f_{l,P}(Q)^{(q^k-1)/l}$ since $l|\#E(\mathbb{F}_q)$ and k is the embedding degree. They also proved that for some supersingular curves with embedding degree $k = 2, 4, 6$ the vertical line evaluation part $\upsilon_{(a+b)P}(Q)$ can be omitted in Miller's algorithm by using a distortion map ψ from $E(\mathbb{F}_q)$ to $E(\mathbb{F}_{q^k})$. Their modified Tate pairing is as follows:

$$\hat{\tau}_l \colon E(\mathbb{F}_q)[l] \times E(\mathbb{F}_q)[l] \to \mu_l \text{ by } (P,Q) \mapsto \tau_l(P, \psi(Q)).$$

It can be directly computed that $f_{N,P}(\psi(Q))^{(q^k-1)/N} = f_{l,P}(\psi(Q))^{(q^k-1)/l}$, where $N = hl$ for some integer h; this computation was first performed by Galbraith *et al.* [12]. Duursma and Lee [14] and Kwon [15] derived the closed formulas of the Tate pairing for characteristic values of three and two, respectively. Their formulas were deduced by computing $f_{N,P}(\psi(Q))^{(q^k-1)/N}$; therefore, they developed efficient Tate pairing algorithms using these formulas. For the case of a characteristic value of two (resp. three), Kwon [15] (resp. Duursma and Lee [14]) used the equation $N = 2^{2m} + 1$ (resp. $N = 3^{3m} + 1$).

To shorten the main loop of the pairing algorithm, Barreto *et al.* [16] defined the η_T pairing for some supersingular curves as follows:

$$\eta_T(P, \psi(Q)) = f_{T,P}(\psi(Q))^W$$

where $T = q \mod l$, $W = (q^k - 1)/N$, N is an integer such that $l|N, N|q^k - 1$, and $T^a - 1 = LN$ for some a, L, and $l \nmid L$. The bilinearity and non-degeneracy of the η_T pairing can be confirmed by the following property (see [16], [17] for more details):

$$\tau_l(P, \psi(Q))^L = \eta_T(P, \psi(Q))^{aT^{a-1}}.$$

Hess *et al.* [17] extended it in a more generic manner to the Ate pairing on non-supersingular elliptic curves.

III. SIDE CHANNEL ATTACKS

SCAs commonly utilize a relation between the side-channel information related to secret data and internal values during cryptographic operations [18], [19]. An attacker utilizes side-channel information such as computation timing, power consumption, and electromagnetic radiation for confirming the accuracy of his or her guess about the secret information. The aim of the attack is to guess the secret value (or some related information) stored at the target device. If an attacker is allowed to observe the side-channel information a few times and is able to directly interpret them, it is termed simple power analysis (SPA). If the attacker can analyze the side-channel information several times using a statistical tool, the process is termed DPA. The standard DPA utilizes a correlation function that can determine whether a specific bit is related to the observed calculation. In particular, if the information about the time taken to execute cryptographic algorithms is utilized, the attack is termed a timing attack (TA). Although SCAs and countermeasures are becoming increasingly well understood, the current emphasis in terms of asymmetric key schemes is mainly on RSA, ECC, and XTR [24].

Recently, new primitives such as pairing algorithms have investigated. First, Page, and Vercauteren proposed fault and SCAs against the Duursma-Lee algorithm [20]. Very recently, Whelan and Scott investigated practical pairing algorithms such as the Tate, Eta, and Ate pairing using correlation power analysis (CPA) [21], and Kim *et al.* investigated the security of the η_T pairing over binary fields in the context of SCAs [22].

A number of countermeasures have been developed to protect pairing algorithms against SCAs [20]–[23]. In [20], the bilinearity of pairing is utilized to hide the secret information. A pairing is calculated as $\tau_l(P,Q) = \tau_l(aP, bQ)^{1/ab}$, where a and b are random values, or as $\tau_l(P,Q) = \tau_l(P, Q+R)/\tau_l(P,R)$, where R is a random point. Note that although the first equation has an additional factor ab in the exponent, it can be eliminated by the careful selection of a and b such that $ab \equiv 1 \pmod{l}$. Scott proposed a very simple concept in which the Miller variable m in the BKLS algorithm [11] is multiplied by a random element that will be eliminated in the final exponentiation [23]. In [21], it is remarked that the random value must not only be multiplied by the Miller variable, but also be multiplied by all intermediate values that comprise the Miller variable in order to obtain an efficient countermeasure. Kim *et al.* introduced efficient and secure algorithms of the η_T pairing using RPC systems for computing the pairing [22].

IV. CONSTRUCTION OF EFFICIENT COUNTERMEASURE OF PAIRING ALGORITHM AGAINST DPA ATTACKS

A. Motivation

In [22], Kim *et al.* proposed the RPC method as protection against a DPA attack on the η_T pairing over the binary field and proved that their method is the fastest method among the existing countermeasures by estimating the computation cost of all proposed methods [20], [23]. Therefore, the RPC method can be a good starting point for the construction of an efficient and secure pairing algorithm. In the RPC-based countermeasure of the pairing algorithm against a DPA attack, since the inputs of the pairing algorithm are two points $P = (\alpha, \beta)$ and $Q = (x, y)$ on the elliptic curve, there are three possible methods to randomize a given pairing algorithm:

- The point P is randomized as a projective coordinate $(\bar{\alpha}, \bar{\beta}, \bar{\gamma}) = (\bar{\gamma}\alpha, \bar{\gamma}\beta, \bar{\gamma})$, where $\bar{\gamma} \in \mathbb{F}_q^*$.
- The point Q is randomized as a projective coordinate $(\bar{x}, \bar{y}, \bar{z}) = (\bar{z}x, \bar{z}y, \bar{z})$, where $\bar{z} \in \mathbb{F}_q^*$.
- Both the points P and Q are randomized simultaneously.

We do not consider the third method because it yields an inefficient algorithm. In order to develop an efficient RPC-based countermeasure against the DPA attack, the following two problems need to be resolved:

- Among the above two RPC methods, the method that yields more efficient RPC-based countermeasure must be selected.
- More extensively, a modification of the original pairing algorithm must be found to reduce the computational cost of the RPC-based countermeasure.

The drastic increase in the computation cost is mainly caused by the modification of the equation over the extension field in the main loop of the pairing algorithm. Therefore, the equation over the extension field needs to be examined in detail. In the next section, we carefully investigate a special equation over the extension field for four inputs in the base field.

B. RPC Method on Balanced Forms

Suppose that $f(\alpha,\beta;x,y)$ is a polynomial over a finite field $\mathbb{F}_q$ for two variable pairs (α,β) and (x,y), and $f_{(\alpha,\beta)}$ denotes the same polynomial rearranged as an (α,β)-variable polynomial. Similarly, $f_{(x,y)}$ denotes $f(\alpha,\beta;x,y)$ rearranged as an (x,y)-variable polynomial.

Definition 1: $f_{(\alpha,\beta)}$ is called a balanced form over $\mathbb{F}_q$ if it is represented as follows:

$$f_{(\alpha,\beta)} := \sum_{i=n}^{1} (h_i(x,y)\alpha^{e_i} + g_i(x,y)\beta^{e_i}) + c_0(x,y) \quad (1)$$

such that

- $e_n > e_{n-1} > \cdots > e_1 \in \mathbb{Q} \setminus \{0\}$ and
- for $i = 1,\cdots,n$, $h_i(x,y)$ and $g_i(x,y)$ are not zeros simultaneously.

Furthermore, if $f_{(\alpha,\beta)}$ and $f_{(x,y)}$ are both balanced forms, then $f(\alpha,\beta;x,y)$ is said to be a *balanced form*.

In Definition 1, $h_i(x,y)$ and $g_i(x,y)$ are called *coefficient polynomials*, n and e_n are called an *index* and *degree* of $f_{(\alpha,\beta)}$, respectively, and each $h_i(x,y)\alpha^{e_i}$ or $g_i(x,y)\beta^{e_i}$ is called a *term* of $f_{(\alpha,\beta)}$. Note that the index and degree are regarded as zero in the case of $f_{(\alpha,\beta)} = 0$ or 1. More explicitly, the index of $f_{(\alpha,\beta)}$ is defined as follows:

Definition 2: Suppose that $f(\alpha,\beta;x,y)$ is a balanced form and $f_{(\alpha,\beta)}$ is represented as shown in (1). Then,

- the index of $f_{(\alpha,\beta)}$ is defined as $n-1$ if the constant term $c_0(x,y)$ is zero or one, and n otherwise.
- The index of $f_{(\alpha,\beta)}$ is defined as the index of $f'_{(\alpha,\beta)}$ if $f_{(\alpha,\beta)} = (f'_{(\alpha,\beta)})^{e'}$, for some integer e', where $f'_{(\alpha,\beta)}$ is also a balanced form.

Definition 3: For a given balanced form $f(\alpha,\beta;x,y)$, an (α,β)-RPC applied form $\hat{f}(\alpha,\beta;\gamma)$ is defined as follows:

$$\sum_{i=n}^{1} \gamma^{e_n-e_i}\,(h_i(x,y)\alpha^{e_i} + g_i(x,y)\beta^{e_i}) + \gamma^{e_n} c_0(x,y).$$

Similarly, we can define the notion of an *(x,y)-RPC applied form* $\hat{f}(x,y;z)$. From the definitions of the RPC applied form and the index, we can directly obtain the following lemma.

Lemma 1: Let $f(\alpha,\beta;x,y)$ be a balanced form. Suppose that I and e are the index and degree of $f_{(\alpha,\beta)}$, respectively, and $\hat{f}(\alpha,\beta;\gamma)$ is an (α,β)-RPC applied form. If α,β,γ,x, and y are regarded as elements in $\mathbb{F}_q^*$, then I field multiplications are additionally required for the computation of $\hat{f}(\alpha,\beta;\gamma)$, compared to the case of $f_{(\alpha,\beta)}$ when the computations of γ^*s are ignored.

Suppose that an extension field $\mathbb{F}_{q^k}$ over $\mathbb{F}_q$ is represented by a polynomial basis $\{1,t,\cdots,t^{k-1}\}$. Now, let us consider a polynomial basis F on the extension field $\mathbb{F}_{q^k}$ such that

$$F = f_0 + f_1 t + f_2 t^2 + \cdots + f_{k-1}t^{k-1} \quad (2)$$

where $f_i(\alpha,\beta;x,y)$ is a balanced form over $\mathbb{F}_q$ for each $i = 0,\cdots,k-1$. Then, F is called a *balanced form over the extension field* $\mathbb{F}_{q^k}$. Let $I_{i(\alpha,\beta)}$ and $I_{i(x,y)}$ be the indices of $f_{i(\alpha,\beta)}$ and $f_{i(x,y)}$ for each $i = 0,\cdots,k-1$, respectively, and let $e_{i(\alpha,\beta)}$ be the degree of $f_{i(\alpha,\beta)}$ for $i = 0,\cdots,k-1$. For convenience, we present several definitions and notations as follows:

- $F_{(\alpha,\beta)}$ (resp. $F_{(x,y)}$) denotes a rearranged equation of F with $f_{i(\alpha,\beta)}$ (resp. $f_{i(x,y)}$) for $i = 0,\cdots,k-1$.
- $\sum_{i=0}^{k-1} I_{i(\alpha,\beta)}$ (resp. $\sum_{i=0}^{k-1} I_{i(x,y)}$) is called an (α,β) (resp. (x,y))-*total index* of F, and it is denoted by $I_{(\alpha,\beta)}$ (resp. $I_{(x,y)}$).
- $\max\{e_{i(\alpha,\beta)} | i = 0,\cdots,k-1\}$ is called an (α,β)-*maximum degree* of F, and it is denoted by $e_{(\alpha,\beta)}$. In a similar manner, we define an (x,y)-maximum degree of F, $e_{(x,y)}$.
- $D_{(\alpha,\beta)}$ denotes the number of elements of $\{e_{i(\alpha,\beta)} | 0 \neq e_{i(\alpha,\beta)} \neq e_{(\alpha,\beta)}$ for $i = 0,\cdots,k-1\}$, and the notation of $D_{(x,y)}$ has a similar meaning with respect to x,y.
- $C_{(\alpha,\beta)}$ (resp. $C_{(x,y)}$) denotes the number of field multiplications for efficiently computing $f_{i(\alpha,\beta)}$'s (resp. $f_{i(x,y)}$'s) for $i = 0,\cdots,k-1$.

Definition 4: An (α,β)-RPC applied form of $F_{(\alpha,\beta)}$, denoted by $\hat{F}_{(\alpha,\beta)}$, is defined as follows:

$$\hat{F}_{(\alpha,\beta)} := \sum_{i=0}^{k-1} \gamma^{e_{(\alpha,\beta)}-e_{i(\alpha,\beta)}} \hat{f}_i(\alpha,\beta;\gamma) t^i.$$

Similarly, an (x,y)-RPC applied form of $F_{(x,y)}$, which is denoted by $\hat{F}_{(x,y)}$, is defined in the same way.

From Lemma 1, we can prove the following theorem for calculating the computation cost of the RPC applied form of F.

Theorem 1: Let F be a balanced form over $\mathbb{F}_{q^k}$ as shown in (2), and $\hat{F}_{(\alpha,\beta)}$ be the (α,β)-RPC applied form. Suppose that for each $i \neq j \in \{0,\cdots,k-1\}$, $f_{i(\alpha,\beta)}$ and $f_{j(\alpha,\beta)}$ have no identical terms. If α,β,γ,x, and y are regarded as the elements in $\mathbb{F}_q^*$, then for computing $\hat{F}_{(\alpha,\beta)}$, the required number of field multiplications is as follows:

$$I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}$$

when the field multiplications for computing γ^*s are ignored.

Proof: By Lemma 1, additional $I_{i(\alpha,\beta)}$ field multiplications are required for computing $\hat{f}_i(\alpha,\beta;\gamma)$ compared to $f_{i(\alpha,\beta)}$ for each $i = 0,\cdots,k-1$, and the number of $\gamma^*\hat{f}_i(\alpha,\beta;\gamma)$'s is exactly equal to $D_{(\alpha,\beta)}$. Therefore $I_{(\alpha,\beta)} + D_{(\alpha,\beta)}$ additional field multiplications are required for the computation of $\hat{F}_{(\alpha,\beta)}$ since $f_{i(\alpha,\beta)}$ and $f_{j(\alpha,\beta)}$ have no identical terms for each $i \neq j \in \{0,\cdots,k-1\}$. Hence, the proof is completed. □

C. Construction of Efficient RPC-Based Countermeasure

Suppose that F is an equation over the extension field in the main loop of a pairing computation algorithm over a given finite field. Let $P = (\alpha, \beta)$ and $Q = (x, y)$ be the input points of the pairing algorithm.

Lemma 2: Assume that F is a balanced form over the extension field. Then, the (α, β) (resp. (x, y))-RPC applied form of F is an equation over the extension field in the main loop of an RPC-based countermeasure randomizing $P = (\alpha, \beta)$ (resp. $Q = (x, y)$).

Proof: The proof is essentially based on the idea in [11], [25]. Since (α, β) is randomized as $(\alpha, \beta, \gamma) \leftarrow (\gamma\alpha, \gamma\beta, \gamma)$ for $\gamma \in \mathbb{F}_q^*$, we apply $\alpha \leftarrow \frac{\alpha}{\gamma}$ and $\beta \leftarrow \frac{\beta}{\gamma}$ on F. Subsequently, F is modified by $\frac{1}{\gamma^{e(\alpha,\beta)}}\hat{F}_{(\alpha,\beta)}$. However, $(\frac{1}{\gamma^{e(\alpha,\beta)}})^{q-1} = 1$ and the final exponent of the pairing has $(q-1)$ as its factor. Therefore, $\frac{1}{\gamma^{e(\alpha,\beta)}}$ can be ignored in the computation of the pairing. □

Lemma 2 and Theorem 1 directly give us the following corollary on the efficiency of the RPC based countermeasure of the pairing algorithm.

Corollary 1: Let $P = (\alpha, \beta)$ and $Q = (x, y)$ be the input points of the pairing algorithm and F be an equation in its the main loop. Suppose that F is a balanced form over the extension field as shown in (2), and $f_{i(\alpha,\beta)}$ and $f_{j(\alpha,\beta)}$ have no identical terms for each $i \neq j \in \{0, \cdots, k-1\}$. Then,

$$I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}$$

field multiplications are required for computing $\hat{F}_{(\alpha,\beta)}$ when field multiplications for computing γ^*s are ignored.

Let F be a balanced form that is the extension field equation in the main loop of a given pairing algorithm. Then, from Corollary 1, $I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}$ is a good tool to measure the efficiency of the RPC based countermeasure of the algorithm since the main computation cost of the pairing algorithm is caused by the field multiplications for computing F. If we select one randomizing point between two input points $P = (\alpha, \beta)$ and $Q = (x, y)$, say P, then $I_{(\alpha,\beta)}$ and $D_{(\alpha,\beta)}$ values are fixed; however, there might be a chance to reduce $C_{(\alpha,\beta)}$ because we can modify the coefficient polynomials of $f_{i(\alpha,\beta)}$ for each $i = 0, \cdots, k-1$. Hence, we can propose the following method for constructing an efficient and secure pairing algorithm from a given pairing algorithm:

Construction of an efficient countermeasure against DPA	
Step 1.	Determine which method is more efficient between the RPC methods using the points P and Q, respectively, by investigating the $(\_,\_)$-total index, $D_{(\_,\_)}$ value, and $C_{(\_,\_)}$ value of the F in the main loop of the algorithm by Corollary 1.
Step 2.	Assume that the method using the point P is selected in the first step. Then, modify each $f_{i(\alpha,\beta)}$ for $i = 0, \cdots, k-1$ to reduce the value $C_{(\alpha,\beta)}$ if possible.
Step 3.	Apply RPC method randomizing the point P on this modified algorithm

In the above construction, if we obtain a new modified pairing algorithm from the given pairing algorithm in steps 1 and 2, the modified algorithm can be called an *RPC-friendly pairing algorithm*, because the modified algorithm yields an efficient RPC-based countermeasure against DPA attacks. The practical application and examples of our construction method are examined in the next section.

V. APPLICATION TO EXISTING PAIRING ALGORITHMS

A. Efficiency of RPC Methods Randomizing Two Input Points Respectively

In this section, we give two examples of the efficiency of RPC-based countermeasures. First, we apply the RPC method to the η_T pairing algorithm [16] on supersingular curves with characteristic two; second, we apply the RPC method to the Tate pairing algorithm [14] with characteristic three.

Algorithm 1 describes Barreto *et al.*'s η_T pairing algorithm over binary fields [16] (for details of the closed formula, see Appendix).

Algorithm 1 η_T pairing algorithm on the curve $E_b : Y^2 + Y = X^3 + X + b$ over $\mathbb{F}_{2^m}$ where $b \in \{0, 1\}$ and m odd [1].

Input: $P = (\alpha, \beta)$ and $Q = (x, y)$.
Output: $\eta_T(P, \psi(Q))$.
$w \leftarrow \alpha + \frac{(m-1)}{2}$
$f \leftarrow w(x+\alpha+1)+y+(\beta+b+\epsilon_{(m+1)/2})+(w+x)s+t$
for $i = 0$ to $(m-1)/2$ **do**
$w \leftarrow \alpha + \frac{(m+1)}{2}$, $\alpha \leftarrow \sqrt{\alpha}$, $\beta \leftarrow \sqrt{\beta}$
$g \leftarrow w(\alpha+x+\frac{(m+1)}{2})+y+(\beta+(1-\frac{(m+1)}{2})\alpha+\epsilon_{(m-1)/2})+(w+x)s+t$
$f \leftarrow fg$
if $i < (m-1)/2$ **then**
$x \leftarrow x^2$, $y \leftarrow y^2$
end if
end for
return $f^W = f^{(2^{2m}-1)(2^m+1-\epsilon 2^{(m+1)/2})}$

The term ϵ_i in steps 2 and 5 of Algorithm 1 is defined as follows (it is also defined in (9) of the Appendix):

$$\epsilon_i = \begin{cases} 0, & \text{if } 0, 1 \mod 4, \\ 1, & \text{otherwise.} \end{cases}$$

In [22], Kim *et al.* randomized the input point Q after a comparison of efficiencies between RPC based countermeasures using input points P and Q, respectively. Now, we can easily examine the efficiency of the RPC methods randomizing P and Q, respectively, by investigating $I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}$ and $I_{(x,y)} + D_{(x,y)} + C_{(x,y)}$ of Corollary 1.

The extension field F in step 5 of Algorithm 1 can be accurately described as follows:

$$F := f_0 + f_1 s + t, \text{ where}$$

$$f_0 = \left(\alpha + \frac{(m+1)}{2}\right)\left(\sqrt{\alpha} + x + \frac{(m+1)}{2}\right) + y + \sqrt{\beta} + \left(1 - \frac{(m+1)}{2}\right)\sqrt{\alpha} + \epsilon_{(m-1)/2} \text{ and}$$

$$f_1 = \alpha + \frac{(m+1)}{2} + x. \tag{3}$$

Example 1: From (3),

$$f_{0(\alpha,\beta)} = \alpha^{3/2} + \left(x + \frac{(m+1)}{2}\right)\alpha + \left(\alpha^{1/2} + \beta^{1/2}\right) + \left(\frac{(m+1)}{2}\left(x + \frac{(m+1)}{2}\right) + y + \epsilon_{(m-1)/2}\right),$$

$$f_{1(\alpha,\beta)} = \alpha + \left(x + \frac{(m+1)}{2}\right),$$

$$f_{0(x,y)} = w(\alpha,\beta)x + y + c(\alpha,\beta), \text{ and}$$

$$f_{1(x,y)} = x + w(\alpha,\beta), \text{ where} \quad (4)$$

$$w(\alpha,\beta) = \left(\alpha + \frac{(m+1)}{2}\right) \text{ and} \quad (5)$$

$$c(\alpha,\beta) = \alpha^{3/2} + \frac{(m+1)}{2}\alpha + \alpha^{1/2} + \beta^{1/2} + \frac{(m+1)}{2} + \epsilon_{(m-1)/2}. \quad (6)$$

Then, the (α,β) (resp. (x,y))-total index of F is 4 (resp. 2), $D_{(\alpha,\beta)} = 1$ and $D_{(x,y)} = 0$. $f_{1(\alpha,\beta)}$ and $f_{1(x,y)}$ do not require any field multiplication. $f_{0(\alpha,\beta)}$ needs two field multiplications (for computing $\alpha^{3/2} = \alpha\alpha^{1/2}$ and $\left(x + \frac{(m+1)}{2}\right)\alpha$). Two field multiplications are also required for $f_{0(x,y)}$ (for computing $w(\alpha,\beta)x$ and $\alpha^{3/2}$). Therefore, $I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)} = 4 + 1 + 2 = 7$ and $I_{(x,y)} + D_{(x,y)} + C_{(x,y)} = 2 + 0 + 2 = 4$. Since base-field squaring is relatively inexpensive [26] and the method in [27] for computing square roots is as fast as squaring, the field multiplication cost is sufficient to compare the efficiencies of the two RPC methods. Hence, the RPC method using the point Q is more efficient than the method using P.

Example 2: In Duursma-Lee's Tate pairing algorithm over fields using characteristic three [14], [15], the equation in the main loop is as follows:

$$F := f_0 - f_1\sigma - f_2\rho - \rho^2$$
$$\text{where } f_0 = -f_2^2,\ f_1 = \beta^3 y,\ f_2 = \alpha^3 + x + b,\ b = \pm 1$$

for given input points $P = (\alpha,\beta)$ and $Q = (x,y)$. Then, the indices of $f_{0(\alpha,\beta)}$ and $f_{2(\alpha,\beta)}$ (resp. $f_{0(x,y)}$ and $f_{2(x,y)}$) are both 1 since $f_0 = -f_2^2$. Therefore $I_{(\alpha,\beta)} = 1+0+1+0 = I_{(x,y)}$. Furthermore, $D_{(\alpha,\beta)} = 2 = D_{(x,y)}$ and $C_{(\alpha,\beta)} = 2 = C_{(x,y)}$. Since the constant term $(x + b)$ (resp. $(\alpha^3 + b)$) of $f_{0(\alpha,\beta)}$ (resp. $f_{0(x,y)}$) is repeated at $f_{2(\alpha,\beta)}$ (resp. $f_{2(x,y)}$), we can reduce the number of field multiplications by one for computing $F_{(\alpha,\beta)}$ (resp. $F_{(x,y)}$). Therefore, $(I_{(\alpha,\beta)} + D_{(\alpha,\beta)} + C_{(\alpha,\beta)}) - 1 = 5$ (resp. $(I_{(x,y)} + D_{(x,y)} + C_{(x,y)}) - 1 = 5$) field multiplications are required for computing $\hat{F}_{(\alpha,\beta)}$ (resp. $\hat{F}_{(x,y)}$). Hence, the RPC methods implemented by using the point P and Q have the same field multiplication cost when the cubing and the cubic root computations are ignored. However, since the cubic root computation is generally more expensive than the cubing computation, we can conclude that the RPC method using the point P is more efficient than the method using Q.

B. RPC-Friendly η_T Pairing Algorithm over Binary Fields

In this section, we develop an RPC-friendly algorithm for the η_T pairing over binary fields using steps 1 and 2 of our construction method.

- Since the RPC-method using the point $Q = (x,y)$ is more efficient than the method using the point P, as shown by Example 1, we first rearrange (3) in the main loop of Algorithm 1 with the (x,y)-variable as observed in (4)–(6).
- Second, we modify the coefficient polynomial $c(\alpha,\beta)$ in (6) as follows:

$$\begin{aligned} c(\alpha,\beta) &= \left(\alpha^3 + \alpha\right)^{1/2} + \frac{(m+1)}{2}\alpha + \beta^{1/2} + \frac{(m+1)}{2} + \epsilon_{(m-1)/2} \text{ by the Weierstrass equation of } E_b, \\ &= \left(\beta^2 + \beta + b\right)^{1/2} + \frac{(m+1)}{2}\alpha + \beta^{1/2} + \frac{(m+1)}{2} + \epsilon_{(m-1)/2} \\ &= \frac{(m+1)}{2}\alpha + \beta + \frac{(m+1)}{2} + b + \epsilon_{(m-1)/2} \\ &= \frac{(m+1)}{2}w(\alpha,\beta) + \beta + b + \epsilon_{(m-1)/2} \end{aligned} \quad (7)$$

  where $w(\alpha,\beta)$ is defined as in (5). Consequently, the modified polynomial $c(\alpha,\beta)$ does not require any field multiplication. Therefore, $C_{(x,y)}$ is reduced by one.

More explicitly (7) shows that

$$g_{2^{-j}P'}(\psi(Q))^{2^j} = w_j x^{(j)} + y^{(j)} + \left(\frac{(m+1)}{2}w_j + \beta^{(-j)} + b + \epsilon_{(m-1)/2}\right) + (w_j + x^{(j)})s + t$$

where $w_j = (\alpha^{(-j)} + \frac{(m+1)}{2})$ (for details of the notations, see the Appendix).

Algorithm 2 RPC-friendly algorithm of η_T pairing on the curve $E_b : Y^2 + Y = X^3 + X + b$ over $\mathbb{F}_{2^m}$ where $b \in \{0,1\}$ and m odd.

Input: $P = (\alpha,\beta)$ and $Q = (x,y)$.
Output: $\eta_T(P,\psi(Q))$.

1: $w \leftarrow \alpha + \frac{(m-1)}{2}$
2: $f \leftarrow w(x+\alpha+1) + y + (\beta + b + \epsilon_{(m+1)/2}) + (w+x)s + t$
3: **for** $i = 0$ to $(m-1)/2$ **do**
4: $w \leftarrow \alpha + \frac{(m+1)}{2}$
5: $g \leftarrow wx + y + (\frac{(m+1)}{2}w + \beta + b + \epsilon_{(m-1)/2}) + (w+x)s + t$
6: $f \leftarrow fg$
7: **if** $i < (m-1)/2$ **then**
8: $\alpha \leftarrow \sqrt{\alpha}$, $\beta \leftarrow \sqrt{\beta}$, $x \leftarrow x^2$, $y \leftarrow y^2$
9: **end if**
10: **end for**
11: **return** $f^W = f^{(2^{2m}-1)(2^m+1-\epsilon 2^{(m+1)/2})}$

Our new RPC-friendly η_T pairing algorithm is shown in Algorithm 2. Algorithm 2 reduces the computational cost by two square root computations on the base field $\mathbb{F}_q$, compared with the Algorithm 1 (i.e., Barreto *et al.*'s algorithm). Therefore, the reduction due to this modification is only a negligible amount of the total computation cost of the η_T pairing. That is, the field multiplication costs of the Algorithms 1 and 2 are the same. Nevertheless, from the viewpoint of the RPC-based countermeasure, this small modification causes a significant difference.

In Algorithm 2, the (x, y)-total index of the equation in the main loop is 2, $D_{(x,y)} = 0$, and $C_{(x,y)} = 1$. Therefore, $I_{(x,y)} + D_{(x,y)} + C_{(x,y)} = 3$ field multiplications are required for computing the equation in the main loop of the RPC-based countermeasure in Algorithm 2 by Corollary 1. However, four field multiplications are required for computing the equation in the main loop of the RPC method (i.e., Kim *et al.*'s algorithm [22]) in the original Algorithm 1 (i.e., Barreto *et al.*'s algorithm [16]) (see Example 1).

Algorithm 3 describes the RPC based countermeasure applied to our RPC-friendly algorithm. The total field multiplication cost of Algorithm 3 is $6(m+1)M+5M$ since the total cost of Kim *et al.* algorithm is $6.5(m+1)M + 5M$ [22], where M denotes one base field multiplication cost.

Since the total cost of Barreto *et al.* algorithm and our RPC-friendly algorithm is $3.5(m+1)M + 1M$ [22], the additional field multiplication cost of Algorithm 3 (resp. Kim *et al.*'s algorithm) is $2.5(m+1)M + 4M$ (resp. $3(m+1)M + 4M$). Consequently, our Algorithm 3 reduces the additional cost by 17% for $m = 239$, as compared with Kim *et al.*'s algorithm [22].

Algorithm 3 Efficient and secure η_T pairing algorithm on the curve $E_b : Y^2 + Y = X^3 + X + b$ over $\mathbb{F}_{2^m}$ where $b \in \{0, 1\}$ and m odd.

Input: $P = (\alpha, \beta)$ and $Q = (x, y)$.
Output: $\eta_T(P, \psi(Q))$.
1: Choose $\bar{z} \in \mathbb{F}_q^*$ at random
2: $\bar{x} \leftarrow \bar{z}x$, $\bar{y} \leftarrow \bar{z}y$
3: $w \leftarrow \alpha + \frac{(m-1)}{2}$
4: $f \leftarrow w(\bar{x} + \bar{z}(\alpha+1)) + \bar{y} + \bar{z}(\beta + b + \epsilon_{(m+1)/2}) + (\bar{z}w + \bar{x})s + \bar{z}t$
5: **for** $i = 0$ to $(m-1)/2$ **do**
6: $w \leftarrow \alpha + \frac{(m+1)}{2}$
7: $g \leftarrow w\bar{x} + \bar{y} + \bar{z}(\frac{(m+1)}{2}w + \beta + b + \epsilon_{(m-1)/2}) + (\bar{z}w + \bar{x})s + \bar{z}t$
8: $f \leftarrow fg$
9: **if** $i < (m-1)/2$ **then**
10: $\alpha \leftarrow \sqrt{\alpha}$, $\beta \leftarrow \sqrt{\beta}$, $\bar{x} \leftarrow \bar{x}^2$, $\bar{y} \leftarrow \bar{y}^2$, $\bar{z} \leftarrow \bar{z}^2$
11: **end if**
12: **end for**
13: **return** $f^W = f^{(2^{2m}-1)(2^m+1-\epsilon 2^{(m+1)/2})}$

VI. CONCLUSION

In this study, we have performed a measurement of the pairing computing algorithm in order to estimate the efficiency of an RPC-based countermeasure against SCAs. We have been able to construct a method to yield an efficient countermeasure of the pairing algorithm against SCAs. Using this method, we have presented an RPC-friendly η_T pairing algorithm over binary fields from the original Barreto *et al.*'s algorithm. The proposed RPC-friendly η_T pairing algorithm reduces the computation cost by two square root computations and has only a slight advantage in efficiency. However, if we apply the RPC method to this algorithm as protection against DPA attacks, then this countermeasure reduces the additional computation cost by 17%, compared with that in the case of application of the RPC method [22] to Barreto *et al.*'s algorithm, which is the most efficient existing countermeasure. This implies that a small modification of the original algorithm might have a significant effect on the efficiency of DPA countermeasures.

REFERENCES

[1] D. Boneh and M. Franklin, "Identity based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, 2003.
[2] R. Sakai and M. Kasahara, "ID based cryptosystems with pairing on elliptic curve," *Cryptography ePrint Archive*, Report 2003/054, 2003. [Online]. Available: http://eprint.iacr.org/2003/054.
[3] J. C. Cha and J. H. Cheon, "An indentity-based signature from gap diffie-Hellman groups," in *Proc. PKC 2003*, LNCS 2567, pp. 18–30.
[4] F. Hess, "Exponent group signature schemes and efficient identity based signature schems based on pairing," in *Proc. SAC 2002*, LNCS 2595, pp. 310–324.
[5] K. G. Paterson, "ID-based signature from pairings on elliptic curves," *Electron. Lett.*, vol. 38, no. 18, pp. 1025–1026, 2002.
[6] A. Joux, "A one round protocol for tripartite diffie-Hellman," *J. Cryptology*, vol. 17, no. 4, pp. 263–276, 2004.
[7] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.
[8] N. P. Smart, "An identity based authentication key agreement protocol based on pairing," *Electron. Lett.*, vol. 38, no. 13, pp. 630–632, 2002.
[9] G. Frey and H. G. Rück, "A remark concening m-divisibility and the discrete logarithm in the divisor class group of curves," *Math. Comput.*, vol. 62, pp. 865–874, 1994.
[10] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Trans. Inf. Theory*, vol. 39, no. 5, pp. 1639–1646, 1993.
[11] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *Proc. CRYPTO 2002*, LNCS 2442, pp. 354–368.
[12] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," in *Proc. ANTS V*, 2002, LNCS 2369, pp. 324–337.
[13] V. S. Miller, "Short programs for functions on curves," unpublished manuscript, 1986.
[14] I. Duursma and H.-S. Lee, "Tate pairing implementation for hyperelliptic curves $y^2 = x^p - x + d$," in *Proc. Asiacrypt 2003*, LNCS 2894, pp. 111–123.
[15] S. Kwon, "Efficient Tate pairing computation for elliptic curves over binary fields," in *Proc. ACISP 2005*, LNCS 3574, pp. 134–145.
[16] P. S. L. M. Barreto, S. Galbraith, C. OhEigeartaigh, and M. Scott, "Efficient pairing computation on supersingular abelian varieties," *Designs, Codes, and Cryptography*, vol. 42, no. 3, pp. 239–271, 2007.
[17] F. Hess, N. Smart, and F. Vercauteren, "The eta pairing revisited," *IEEE Trans. Inf. Theory*, vol. 52 no. 10, pp. 4595–4602, 2006.
[18] P. Kocher, "Timing attacks on implementations of diffie-Hellman, RSA, DSS, and other systems," in *Proc. CRYPTO 1996*, LNCS 1109, pp. 104–113.
[19] C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. CRYPTO 1999*, LNCS 1666, pp. 388–397.
[20] D. Page and F. Vercauteren, "Fault and sde-channel attacks on pairing based cryptography," *Cryptology ePrint Archive*, Report 2004/283, 2005. [Online]. Available: http://eprint.iacr.org/2004/283.
[21] C. Whelan and M. Scott, "Side channel analysis of practical pairing implementations: Which path is more secure?" *Cryptography ePrint Archive*, Report 2006/237, 2006. [Online]. Available: http://eprint.iacr.org/2006/237.
[22] T. H. Kim, T. Takagi, D.-G. Han, H. W. Kim, and J. Lim, "Side channel attacks and countermeasures on pairing based cyrptosystems over binary fields," in *Proc. CANS 2006*, LNCS 4301, pp. 168–181.
[23] M. Scott, "Computing the Tate pairing," in *Proc. CT-RSA 2005*, LNCS 3376, pp. 293–304.
[24] A. K. Lenstra and E. R. Verheul, "The XTR public key system," in *Proc. CRYPTO 2000*, LNCS 1880, pp. 1–19.
[25] P. S. L. M. Barreto, B. Lynn, and M. Scott, "On the selection of pairing-friendly groups," in *Proc. SAC 2003*, LNCS 3006, pp. 17–25.
[26] D. Hankerson, J. L. Hernandez, and A. Menezes, "Software implementation of elliptic curve cryptography over binary fields," in *Proc. CHES 2000*, LNCS 1965, pp. 1–24.
[27] K. Fong, D. Hankerson, J. López, and A. Menezes, "Field inversion and point halving revisited," *Technical Report CORR 2003-18*, University of Waterloo, Aug. 2002.
[28] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[29] D. Page and F. Vercauteren, "A fault attack on pairing based cryptography," *IEEE Trans. Comput.*, vol. 55, no. 9, pp. 1075–1080, 2006.

APPENDIX

A. Pairing Computation over Binary Fields

In this Appendix, we briefly review the closed formula of the Tate and η_T pairing over binary fields in [15], [16]. We consider the elliptic curves over binary fields $\mathbb{F}_q$, where $q = 2^m$ and m is odd, as follows:

$$E_b : Y^2 + Y = X^3 + X + b \text{ where } b \in \{0, 1\}.$$

Then, E_b has the embedding degree $k = 4$ [15], [16], [28] and $\#E_b(\mathbb{F}_q) = 2^m + 1 + \epsilon 2^{\frac{m+1}{2}}$, where

$$\epsilon = \begin{cases} -1, & \text{if } (m = 1, 7 \pmod 8 \text{ and } b = 1) \text{ or} \\ & (m = 3, 5 \pmod 8 \text{ and } b = 0), \\ 1, & \text{otherwise.} \end{cases}$$

In the elliptic curve E_b, the extension field $\mathbb{F}_{q^4}$ is represented by the basis $\{1, s, t, st\}$ such that $s^2 + s + 1 = 0$ and $t^2 + t + s = 0$. The distortion map is given by $\psi(x, y) = (x + s^2, y + xs + t)$. Furthermore, in this setting, Barreto *et al.* [16] proved that the T value of the η_T pairing is $2^{(m+1)/2} + \epsilon$.

It can be directly induced that for a given $P = (\alpha, \beta)$,

$$2^i P = (\alpha_i^{(2i)}, \beta_i^{(2i)})$$
$$\text{where } (x_i, y_i) = \phi^i(x, y),\ \phi(x, y) = (x + 1, y + x + 1).$$

In the above equation, $\alpha^{(j)}$ (resp. $\beta^{(j)}$) is defined as $\alpha^{(j)} = \alpha^{2^j}$ (resp. $\beta^{(j)} = \beta^{2^j}$) (for further details, see [15], [16]). Then,

$$\phi^i(x, y) = (x + i, y + ix + \epsilon_i), \text{ where} \tag{8}$$

$$\epsilon_i = \begin{cases} 0, & \text{if } 0, 1 \mod 4, \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

In [15], Kwon derived a closed formula of the Tate pairing, and Barreto *et al.* [16] independently found a closed formula of the η_T pairing on the elliptic curve E_b. The following theorem is a summary of these results.

Theorem 2 ([15], [16]) For given $P = (\alpha, \beta), Q = (x, y)$ in $E_b(\mathbb{F}_q)$,

- The Tate pairing $\tau_l(P, \psi(Q)) =$

$$\left(\prod_{i=0}^{m-1} g_{2^i P}(\psi(Q))^{2^{2m-i}} \right)^{2^{2m}-1}$$

where $g_R(X, Y)$ is an equation of the tangent line at R.

- The η_T pairing $\eta_T(P, \psi(Q)) =$

$$\left(\left(\prod_{j=0}^{(m-1)/2} g_{2^{-j}P'}(\psi(Q))^{2^j} \right) \ell(\psi(Q)) \right)^W$$

where $P' = 2^{(m-1)/2} P$, $\ell(X, Y)$ is the equation of a line passing through $2^{m+1/2} P$ and ϵP, and $W = q^k - 1/N = (2^{2m} - 1)(2^m + 1 - \epsilon 2^{m+1/2})$, $N = \#E_b(\mathbb{F}_q)$.

Furthermore,

$$g_{2^i P}(\psi(Q)) = (\alpha_i^{(2i+1)} + 1)(x + 1) + y + \beta_i^{(2i+1)} + b + (\alpha_i^{(2i+1)} + 1 + x)s + t. \tag{10}$$

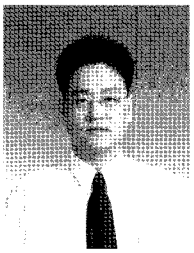
From (8) and (10), it can be directly proved [15] that

$$\begin{aligned} g_{2^i P}(\psi(Q))^{2^{2m-i}} =& (\alpha^{(i+1)} + 1)x^{(-i)} + y^{(-i)} \\ &+ (\alpha^{(i+1)} + \beta^{(i+1)} + b) \\ &+ ((\alpha^{(i+1)} + 1) + x^{(-i)})s + t. \end{aligned}$$

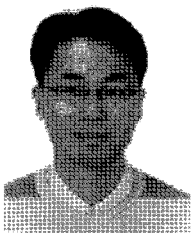
Barreto *et al.* [16] computed that

$$\begin{aligned} g_{2^{-j}P'}(\psi Q)^{2^j} =& w_j(\alpha^{(-1-j)} + x^{(j)} + \frac{(m+1)}{2}) + y^{(j)} \\ &+ \beta^{(-1-j)} + (1 - \frac{(m+1)}{2})\alpha^{(-1-j)} \\ &+ \epsilon_{(m-1)/2} + (w_j + x^{(j)})s + t, \\ \ell(\psi(Q)) =& (\alpha + (m-1)/2)(\alpha + x + 1) + y \\ &+ (\beta + b + \epsilon_{(m+1)/2}) \\ &+ ((\alpha + (m-1)/2) + x)s + t \end{aligned}$$

where $w_j = (\alpha^{(-j)} + \frac{(m+1)}{2})$.



Dooho Choi received his B.S. degree in mathematics from Sungkyunkwan University, Seoul, Korea in 1994, and the M.S. and Ph.D. degrees in mathematics from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 1996 and 2002, respectively. He is currently a Senior Researcher in Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from Jan. 2002. His research interests include security technologies of RFID and sensor network. He is an Editor of the ITU-T X. nidsec-1.



Dongguk Han received his B.S. degree in mathematics from Korea University in 1999, and his M.S. degrees in mathematics from Korea University in 2002. He received Ph.D. of engineering in Information Security from Korea University in 2005. He was a Post-Doc. in Future University-Hakodate, Japan. After finishing the doctor course, he had been an exchange student in Department of Computer Science and Communication Engineering in Kyushu University from Apr. 2004 to Mar. 2005. Now, he is a Senior Researcher in Electronics and Telecommunications Research Institute (ETRI) from Jun. 2006. He is a Member of KIISC, IEEK, and IACR.

Howon Kim received his B.S.E.E. degree from Kyungpook National University, Daegu, Korea, in 1993, and the M.S. and Ph.D. degrees in Electronic and Electrical Engineering from Pohang University of Science and Technology (POSTECH), Pohang, Korea, in 1995 and 1999, respectively. From July 2003 to June 2004, he studied at the COSY group at the Ruhr-University of Bochum, Germany. He was a Senior Member of technical staff at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. Currently, he works at the Department of Computer Engineering at Pusan National University, Busan, Korea. His research interests include RFID technology, sensor network, information security, and computer architecture. Currently, his main research focus is on mobile RFID technology, sensor network, public key cryptosystem, and its security issues. He is a Member of the IEEE, IEEE Computer Society, and IACR. He is also an Editor of the ISO 24791-6.