Collision Attack on XTR and a Countermeasure with a Fixed Pattern*

Dong-Guk Han[1], Tsuyoshi Takagi[1], Tae Hyun Kim[2],
Ho Won Kim[3], and Kyo Il Chung[3]

[1] Future University-Hakodate, Japan
{christa, takagi}@fun.ac.jp
[2] Center for Information and Security Technologies(CIST),
Korea University, Seoul, Korea
thkim@cist.korea.ac.kr
[3] Electronics and Telecommunications Research Institute(ETRI), Korea
{khw, kyoil}@etri.re.kr

Abstract. Recently, XTR is considered as one of good candidates for more energy efficient cryptosystems. Among the family of XTR algorithms, the Improved XTR Single Exponentiation (XTR-ISE) is the most efficient one suitable for ubiquitous computer. Even though the security of such devices against side channel attacks is very dangerous, there are few works on side channel attacks against XTR-ISE. In this paper we propose a new collision attack on XTR-ISE. The analysis complexity of the proposed one is about 2^{40} where the key size is 160-bit, which is 55% improvement from the previously best known analysis of Page-Stam. We also propose a novel countermeasure using a fixed pattern which is secure against SPA. In the sense of both efficiency and security the proposed countermeasure is the best one among the previous countermeasures- it is about 30% faster.

Keywords: Ubiquitous computer, XTR public key system, XTR Exponentiation Algorithms, Side Channel Attacks, Collision Attack.

1 Introduction

We are standing to the beginning of the ubiquitous computing era. It is expected that we can accomplish lucrative applications by effectively synthesizing the ubiquitous computer with cryptography. The ubiquitous computer only has scarce computational resources (like Smart cards, RFID, Sensor Network), so that we have to make an effort to optimize the memory and efficiency of the security system. Currently there are a few implementations on ubiquitous environments with PKC. In ESAS 2004 Gaubatz-Kaps-Sunar showed an implementation of Rabin and Ntru in sensor networks [6]. Recently Watro et al. showed RSA (in the case the encryption key is 3) is feasible to the applications of ubiquitous computer, and remarked that XTR is one of good candidates for light

* The full version of this paper was posted in the Cryptology ePrint Archive [9].

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 864–873, 2005.

weight cryptosystems in SASN 2004 [18]. However, the applications of ubiquitous computer will be carried into and used in hostile environments and often house sensitive information, for example identity related tokens or financial information, the threat of attack is significant. This threat is magnified by both the potential pay-off and level of anonymity that side channel attacks (SCA) allow [10, 11]. The fact that one can attack a device somewhat remotely via timing and power consumption means that most ubiquitous computing devices need to be aware of similar problems in their operational environments.

In Crypto 2000 Lenstra-Verheul introduced XTR public key cryptosystems [12]. Given the current state of affairs in breaking the discrete logarithm problems over either finite fields or elliptic curves, XTR can compete with elliptic curve cryptosystems (ECC) in terms of both speed and bandwidth. This makes XTR suitable for deployment on similar sorts of constrained devices such as smart-cards, where computational power and storage capacity are both very limited. Among the family of XTR exponentiation algorithms, the Improved XTR Single Exponentiation (XTR-ISE) is the most efficient one suitable for smart-cards, where computational power and memory capacity are both very limited. Even though the security of such devices against side channel attacks is very dangerous, however, there are few works on side channel attacks against XTR-ISE.

In 2004 Chung-Hasan [2] and Page-Stam [14] proposed simple power analysis (SPA) against XTR-ISE and that it was the first try to analyze it with SCA. Chung-Hasan showed it takes 2^{100} tries for an attacker until he/she correctly finds the secret key in XTR-ISE with 160-bits key length. On the other hand, Page-Stam showed it requires 2^{88} tries. However, these results are far worse than well-known square-root type algorithms (Baby-Step-Giant-Step or Pollards' Rho methods).

In this paper we find a new analysis technique, called as XTR collision attack, derived from the structural properties of XTR-ISE. The complexity of XTR collision attack is about $2^{0.25 \cdot l}$ where l is the length of the key, which is about 55% improvement from the result of Page-Stam [14]. Also we propose a novel countermeasure using a fixed pattern which is secure against SPA. In the sense of both efficiency and security the proposed countermeasure is the best one among the previous countermeasures- it is about 30% faster.

2 XTR Exponentiation Algorithm

In this section, we review the fundamental algorithms to calculate traces of powers [12, 15]. For an element $h \in \mathbf{F}_{p^6}^*$ its trace $\mathrm{Tr}(h)$ over $\mathbf{F}_{p^2}$ is defined as a sum of the conjugates over $\mathbf{F}_{p^2}$ of h: $\mathrm{Tr}(h) = h + h^{p^2} + h^{p^4} \in \mathbf{F}_{p^2}$. Throughout this paper, c_n denotes $\mathrm{Tr}(g^n) \in \mathbf{F}_{p^2}$, for some fixed p and g of order q, where q divides $p^2 - p + 1$. Note that $c_0 = 3$ and $c_1 = c$.

An efficient computation of c_n for given p, q and c depends on the recurrence relations $c_{u+v} = c_u c_v - c_v^p c_{u-v} + c_{u-2v}$, and $c_{2u} = c_u^2 - 2c_u^p$, which is derived from the previous one when $u = v$.

By using above two formulae, we define the following two functions called as XTR addition and XTR doubling respectively;

$$A[x, y, z, w] = x \cdot y - y^p \cdot z + w,$$
$$D[x] = x^2 - 2x^p.$$

By using above defined notations we introduce Improved XTR exponentiation algorithms proposed by Stam-Lenstra [15]. The goal of these algorithms is to compute c_n for given c_1 and $n \in Z$, i.e. to compute $Tr(g^n)$ with $Tr(g)$ and an integer n.

Improved XTR Single Exponentiation (XTR-ISE) [15]

Input: c_1 and n where $n > 2$
Output: c_n

1. Initialization:
   1.1. Let $a = \text{round}(\frac{3-\sqrt{5}}{2}n)$ and $b = n - a$ (where round(x) is the integer closest to x).
   1.2. Let $f = 0$. As long as a and b are both even, replace (a, b) by $(a/2, b/2)$ and f by $f + 1$.
   1.3. Let $i = 1$ and $G_i := (Q_0, Q_1, Q_2, Q_3) = (c_1, c_1, 3, c_1^p)$.
2. As long as $a \neq b$
   2.1. If $b > a$
   X_1. if $b \leq 4a$, then $(a, b) \leftarrow (b - a, a)$
   $T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3]$, $T_1 \leftarrow Q_0$,
   $T_2 \leftarrow Q_1$, $T_3 \leftarrow Q_2^p$.
   X_2. else if b is even, then $(a, b) \leftarrow (a, b/2)$
   $T_0 \leftarrow D[Q_0]$, $T_1 \leftarrow Q_1$,
   $T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p]$, $T_3 \leftarrow D[Q_2]$.
   X_3. else if a is odd, then $(a, b) \leftarrow (a, (b - a)/2)$
   $T_0 \leftarrow D[Q_0]$, $T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3]$,
   $T_2 \leftarrow Q_2$, $T_3 \leftarrow D[Q_1]^p$.
   X_4. else (a is even), then $(a, b) \leftarrow (b, a/2)$
   $T_0 \leftarrow D[Q_1]$, $T_1 \leftarrow Q_0$,
   $T_2 \leftarrow Q_3^p$, $T_3 \leftarrow D[Q_2]^p$.
   2.2. Else (if $a > b$)
   Y_1. if $a \leq 4b$, then $(a, b) \leftarrow (a - b, b)$
   $T_0 \leftarrow A[Q_0, Q_1, Q_2, Q_3]$, $T_1 \leftarrow Q_1$,
   $T_2 \leftarrow Q_0$, $T_3 \leftarrow Q_2$.
   Y_2. else if a is even, then $(a, b) \leftarrow (b, a/2)$
   $T_0 \leftarrow D[Q_1]$, $T_1 \leftarrow Q_0$,
   $T_2 \leftarrow Q_3^p$, $T_3 \leftarrow D[Q_2]^p$.
   Y_3. else if b is odd, then $(a, b) \leftarrow (b, (a - b)/2)$
   $T_0 \leftarrow D[Q_1]$, $T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3]$,
   $T_2 \leftarrow Q_2^p$, $T_3 \leftarrow D[Q_0]^p$.
   Y_4. else (b is even), then $(a, b) \leftarrow (a, b/2)$
   $T_0 \leftarrow D[Q_0]$, $T_1 \leftarrow Q_1$,
   $T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p]$, $T_3 \leftarrow D[Q_2]$
   2.3. $i \leftarrow i + 1$ and set $G_i = (T_0, T_1, T_2, T_3)$.
3. Compute $\tilde{c} = A[Q_0, Q_1, Q_2, Q_3] = c_{u+v}$.
4. Output $\tilde{c}_{2f}$.
5. If $a = 1$ then return $\tilde{c}_{2f}$
   else run Improved XTR Single Exponentiation with $c = \tilde{c}_{2f}$ and $n = a$.

3 New Collision Attack on XTR

In this section we find a new analysis technique, called as XTR collision attack, derived from the structural properties of XTR-ISE.

3.1 Some Properties of XTR-ISE

In XTR-ISE, Step 2 consists of eight states X_i and Y_i where $1 \leq i \leq 4$. One state is only determined by the condition of a and b. From XTR-ISE we can derive the following finite markov chain depicted in figure 1.

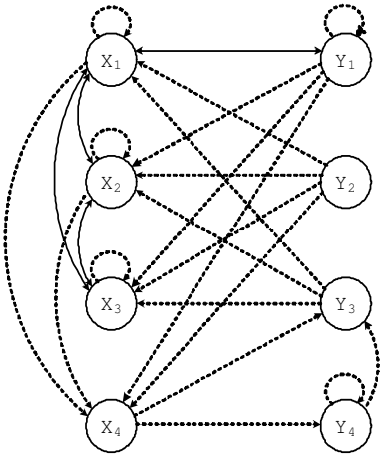

Fig. 1. The finite Markov chain associated with XTR-ISE

Property 1. State Y_2 never occurs in the process of XTR-ISE except the first time.

3.2 Assumptions and Notations

We first introduce some reasonable assumptions which are used in a new attack.

1. $A[x, y, z, w]$ and $D[x]$ are distinguishable by a single measurement of power consumption, whereas $D[x]^p$ and $D[x]$, and $A[x, y, z, w^p]$ and $A[x, y, z, w]$ are indistinguishable, respectively. Here, $x, y, z, w \in \mathbf{F}_{p^2}$.
2. When $\{A[\cdot], D[\cdot], D[\cdot]\}$ are all operated, e.g. in the case of X_3 in XTR-ISE, we assume these three functions are operated according to the following order $A[\cdot]D[\cdot]D[\cdot]$. In more detail, states X_i and Y_i are updated according to the following orders;
   (a) In X_2 and Y_4: the computation order is $T_2 \rightarrow T_0 \rightarrow T_3 \rightarrow T_1$,
   (b) In X_3 and Y_3: the computation order is $T_1 \rightarrow T_0 \rightarrow T_3 \rightarrow T_2$.
3. If $D[c_u]$ and $D[c_v]$ are computed, the attacker is not able to guess the value of c_u nor c_v but he/she is able to check if $c_u = c_v$.

As the required computing time of $A[\cdot]$ is two times of that of $D[\cdot]$ and p-th powering is free (refer to [12]) in XTR, the above Assumption 1 is reasonable. Assumption 3 is also reasonable since this kind of computation usually takes

many clock cycles and depends greatly on the value of the operand. This kind assumption has been used in a stronger variant and validated by Schramm et al. [16] who are able to distinguish collisions during one DES round computation. It was extended to ECC by Fouque at al [5].

Notations: For simplicity, $A[x, y, z, w]$ and $D[x]$ are referred to as **A** and **D**, respectively. Let $S[c_1, n]$ be an AD sequence when the inputs are c_1 and n in XTR-ISE, i.e. **A** and **D** are elements of $S[c_1, n]$, which are written with time-increasing from left to right. Due to the above Assumption 1, **A** and **D** also denote $A[x, y, z, w^p]$ and $D[x]^p$, respectively.

As described in XTR-ISE algorithm, $G_i = (Q_0, Q_1, Q_2, Q_3)$ is the i-th updated intermediate values of $\{Q_j\}_{0 \le j \le 3}$ in Step 2 of XTR-ISE for $i \ge 1$. For $G_i \xrightarrow{T_i} G_{i+1}$ where $T_i \in \{X_j, Y_j\}_{1 \le j \le 4}$, if T_i is one of $\{X_2, X_3, X_4, Y_3, Y_4\}$, then **DD** is computed. We denote these two **DD** as $\mathbf{D_i^1 D_i^2}$ ($\mathbf{D_i^1 D_i^2}$ are carried along for expository purposely only).

3.3 Attacker's Goal

In XTR-ISE, Step 2 consists of eight states X_i and Y_i where $1 \le i \le 4$. One state is only determined by the condition of two integers a and b. However, if an attacker can decide the states executed during the computation then the secret key can be easily reconstructed from the recovered state.

Under Assumption 1, an attacker is able to distinguish **A**, **DD**, and **ADD**. With this information he/she can categorize seven states of XTR-ISE into the following three groups;

- **A** is corresponding to X_1 or Y_1,
- **DD** is corresponding to X_4,
- **ADD** is corresponding to X_2, X_3, Y_3 or Y_4.

However, there are some ambiguity decisions such as (1) X_1 and Y_1 are not distinguished, (2) if **ADD** is observed in AD sequence then there are two possibilities; **ADD** and **A**|**DD**. Using a brute force search technique, one might test around 6 candidates; i.e. **ADD** is corresponding to one of $\{X_2, X_3, Y_3, Y_4, X_1|X_4, Y_1|X_4\}$.

Thus the attacker needs to check the possible candidates until he/she has found the correct one, so in order to improve the efficiency of the attack we want to minimize the number of candidates.

3.4 Analysis Based on the Finite Markov Chain

First we consider the following three types of AD sequences;

- **ADD**|**DD**.
- $\overbrace{\mathbf{ADD}|\mathbf{ADD}|\ldots|\mathbf{ADD}}^{m-times}$, briefly it is denoted as $\{\mathbf{ADD}\}^{\mathbf{m}}$.
- $\mathbf{ADD}|\overbrace{\mathbf{A}\ldots\mathbf{A}}^{m-times}|\mathbf{ADD}$, denoted as $\mathbf{ADD}|\{\mathbf{A}\}^{\mathbf{m}}|\mathbf{ADD}$.

When ADD|DD is observed in AD sequence we can decide $\underbrace{\mathbf{ADD}}_{X_2}|\underbrace{\mathbf{DD}}_{X_4}$. Because the last two **DD** originates from X_4 and the possible preconditions of X_4 are X_1, X_2, and Y_1. Thus **ADD** implies X_2

When $\{\mathbf{ADD}\}^{\mathbf{m}}$ is observed in AD sequence there are 6^m possible combinations from $\{X_2, X_3, Y_3, Y_4, X_1|X_4, Y_1|X_4\}$. However, if we consider the finite markov chain (Fig. 1) then we can reduce the possible number of combinations such as 15 and 39 combinations when m is 2 and 3, respectively. If $m \geq 4$ then the number of all possible combinations from the finite markov chain is $\#[\{\mathbf{ADD}\}^{\mathbf{m}}] = (39m + 48) \cdot 2^{m-5}$.

When ADD|$\{$A$\}^{\mathbf{m}}$|ADD is observed in AD sequence there are $6 \cdot 2^m \cdot 6 = 9 \cdot 2^{m+2}$ combinations of XTR states. However, if we consider the finite markov chain (Fig. 1) then the number of possible combinations is $3 \cdot 2^{m+1}$. Furthermore, we propose the following decision rule derived from the finite markov chain (Fig. 1).

Property 2. If **AADDAA** is observed in AD sequence then we can decide $\mathbf{A}|\underbrace{\mathbf{ADD}}_{\mathbf{x_2 \text{ or } x_3}}|\underbrace{\mathbf{A}}_{\mathbf{x_1}}|\mathbf{A}$.

3.5 XTR Collision Attack

At the previous section, the number of possible combinations for $\{\mathbf{ADD}\}^{\mathbf{m}}$ and **ADD**|$\{$A$\}^{\mathbf{m}}$|**ADD** is decreased by using the finite markov chain of XTR-ISE. In this section, in order to reduce the search space from the finite markov chain we introduce a new attack mainly based on the above assumptions, especially Assumption 3, described in 3.2.

Key Observation: If we focus on **D** operation, we notice that some of them manipulate the same operand. We consider two AD sequences $S[c_1, n]$ and $S[c_2, n]$.

In the case of $\{\mathbf{ADD}\}^{\mathbf{m}}$: For simplicity, we assume $m = 2$, i.e. **ADDADD** is considered. Note that there are 15 combinations of states.

$$S[c_1, n] = \ldots \mathbf{AD_i^1 D_i^2 AD_j^1 D_j^2} \ldots$$
$$S[c_2, n] = \ldots \mathbf{AD_i^1 D_i^2 AD_j^1 D_j^2} \ldots$$

Depending on the combination type, we can observe the following results;

CASE_I: $\mathbf{D_j^1}$ of $S[c_1, n]$ is same to $\mathbf{D_i^1}$ of $S[c_2, n]$,
CASE_II: $\mathbf{D_j^2}$ of $S[c_1, n]$ is same to $\mathbf{D_i^1}$ of $S[c_2, n]$.

According to the above observation, the 15 combination pairs are categorized as

CASE_I: $(X_2, X_2), (X_2, X_3), (X_3, X_2), (X_3, X_3), (X_1, X_4, Y_4), (Y_1, X_4, Y_4), (Y_3, X_2), (Y_3, X_3), (Y_4, Y_4), (X_2, X_1, X_4), (X_3, X_1, X_4), (Y_3, X_1, X_4)$,
CASE_II: $(X_1, X_4, Y_3), (Y_1, X_4, Y_3), (Y_4, Y_3)$.

With this collision information, we can make the following comparison table.

# of all possible combinations			
m	From Exhaustive Search	From the Finite Markov Chain	From Collision Attack
1	6	6	6
2	36	15	10.2
3	216	39	17.77
⋮	⋮	⋮	⋮

From the results of the above table we can see that the average number of trial tests with collision information is drastically decreased compared to that of the finite markov chain.

In the case of ADD|{A}$^{\mathbf{m}}$|ADD: Consider

$$S[c_1, n] = \ldots \mathbf{AD_i^1 D_i^2 \{A\}^m AD_j^1 D_j^2} \ldots$$
$$S[c_2, n] = \ldots \mathbf{AD_i^1 D_i^2 \{A\}^m AD_j^1 D_j^2} \ldots$$

Similar to the previous analysis, we can observe the following results depending on the combination type;

CASE_0: There is no relation between **D** operation of $S[c_1, n]$ and $S[c_2, n]$,
CASE_I: $\mathbf{D_j^1}$ of $S[c_1, n]$ is same to $\mathbf{D_i^1}$ of $S[c_1, n]$,
CASE_II: $\mathbf{D_j^2}$ of $S[c_1, n]$ is same to $\mathbf{D_i^1}$ of $S[c_2, n]$.

The results of the following table show the improvement of analysis complexity.

# of all possible combinations			
m	From Exhaustive Search	From the Finite Markov Chain	From Collision Attack
1	72	12	4.5
2	144	24	9.75
3	288	48	28.87
⋮	⋮	⋮	⋮

Implementation Results: From these classifications, we can reduce the search space order required to detect the whole secret value. From our implementation results the average number of trial XTR exponentiations is roughly given by $2^{0.25 \cdot l}$ where l is the length of the exponents. Thus the complexity of XTR collision attack against XTR-ISE is about 2^{40} where the key length is 160-bit, which is about 55% improvement from the result of Page-Stam [14].

4 Proposed Countermeasure

In this section we explain the proposed algorithm. We modify XTR-ISE to be secure against SCA. The main idea is same to that of Okeya-Takagi scheme [13] for elliptic curve cryptosystems. In XTR-ISE there are three different patterns, **A**, **DD**, and **ADD**. For example, if X_1, Y_1, and X_4 are consecutively operated then the sequence is "**AAADD**", which is no longer the fixed pattern.

We try to generate a XTR operation sequence that has a fixed pattern such that |**ADD**|**ADD**|...|**ADD**|.

Fixed Pattern XTR Single Exponentiation (XTR-FSE)

Input: c_1 and n where $n > 2$
Output: c_n

1. Initialization:
   1.1. Select a random number a in $[1, n-1]$ and $b = n - a$. If a is even, then let $a \leftarrow a + 1$, $b \leftarrow b - 1$.
   1.2. Let $Q_0 = c$, $Q_1 = c$, $Q_2 = 3$, and $Q_3 = c^p$.
2. As long as $a \neq b$
   2.1. If $b > a$
   $\mathcal{X}_1$. if b is even, then $(a, b) \leftarrow (a, b/2)$
   $T_0 \leftarrow D[Q_0]$, $T_1 \leftarrow Q_1$,
   $T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p]$, $T_3 \leftarrow D[Q_2]$.
   $\mathcal{X}_2$. else (b is odd), then $(a, b) \leftarrow (a, (b-a)/2)$
   $T_0 \leftarrow D[Q_0]$, $T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3]$,
   $T_2 \leftarrow Q_2$, $T_3 \leftarrow D[Q_1]^p$.
   2.2. Else (if $a > b$)
   $\mathcal{Y}_1$. if b is odd, then $(a, b) \leftarrow (b, (a-b)/2)$
   $T_0 \leftarrow D[Q_1]$, $T_1 \leftarrow A[Q_0, Q_1, Q_2, Q_3]$,
   $T_2 \leftarrow Q_2^p$, $T_3 \leftarrow D[Q_0]^p$.
   $\mathcal{Y}_2$. else (b is even), then $(a, b) \leftarrow (a, b/2)$
   $T_0 \leftarrow D[Q_0]$, $T_1 \leftarrow Q_1$,
   $T_2 \leftarrow A[Q_0, Q_2, Q_1, Q_3^p]$, $T_3 \leftarrow D[Q_2]$
   2.3. Set $Q_0 \leftarrow T_0$, $Q_1 \leftarrow T_1$, $Q_2 \leftarrow T_2$, $Q_3 \leftarrow T_3$.
3. Compute $\tilde{c} = A[Q_0, Q_1, Q_2, Q_3] = c_{u+v}$.
4. If $a = 1$ then return $\tilde{c}$,
   else goto Step 1. with $c = \tilde{c}$ and $n = a$.

We can prove the following proposition about the efficiency of XTR-FSE.

Proposition 1. *For a given integer n, the proposed algorithm takes on average $1.41 \log_2 n$ iterations in Step 2. Thus the trace value c_n can on average be computed in about $11.2 \log_2(n)$ multiplications in $\mathbf{F}_p$ because each step requires 8 multiplications in $\mathbf{F}_p$ [12].*

4.1 Security Analysis

In this section we discuss the security of the proposed scheme against SPA and DPA.

SPA: As we mentioned in the previous section, the proposed method compute XTR single exponentiation through the fixed pattern $|\mathbf{ADD}|\mathbf{ADD}|\ldots|\mathbf{ADD}|$. The attacker could distinguish XTR operations $D[\cdot]$ and $A[\cdot]$ in XTR-FSE by measurement of the power consumption, but he/she obtains only the identical **ADD** sequence for any input c and n. Therefore, he/she cannot detect the secret scalar n by using SPA.

DPA: The use of scalar randomization such as exponent splitting [3] prevents against DPA. Note that the idea of splitting the data was already abstracted in [4] as a general countermeasure against DPA. The proposed method is using exponent splitting technique as a DPA countermeasure, i.e. we split the input integer n into two parts by picking a random $a \in [1, n-1]$ and rewriting the integer n as $a + (n - a)$. Thus XTR-FSE is secure against DPA.

4.2 Comparison of Empirical Performance and Type of Countermeasure

In this section we compare the computational cost and the type of countermeasures between the proposed countermeasure and the previous ones.

The compared three methods use the exponent splitting method as DPA countermeasure. But the utilized SPA countermeasure is different each others. The countermeasure of ICICS'04 is based on XTR-SE. Their method does not require SPA countermeasure because XTR-SE has the fixed operations **ADD**. On the other hand, the countermeasure of SAC'04 and the proposed method is based on XTR-ISE, which does not has fixed operations. In order to solve this problem Page-Stam proposed the indistinguishable arithmetic with dummy operation sometimes, but the security of indistinguishable arithmetic [17] and the dummy method [19] are recently very controversial. From the result of Table 1 our proposed countermeasure is the best one in XTR in the sense of both efficiency and security.

Table 1. Comparison of empirical performance and type of countermeasure

	Efficiency	Type of Countermeasure	
	Compute $Tr(g^n)$	SPA	DPA
ICICS'04 [8]	$16\log_2(n)$	Fixed Pattern + No Dummy Operation	Exponent Splitting
SAC'04 [14]	$8.5\log_2(n)$	Indistinguishable Assumption + Dummy Operation	Exponent Splitting
Proposed Method	$11.2\log_2(n)$	Fixed Pattern + No Dummy Operation	Exponent Splitting

Acknowledgements

Dong-Guk Han was supported by the Korea Research Foundation Grant. (KRF-2005-214-C00016) and Tae Hyun Kim was supported in part by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

References

1. M. Ciet and C. Giraud, *Transient Fault Induction Attacks on XTR*, Information and Communications Security (ICICS 2004), LNCS 3269, (2004), 440-451.
2. J. Chung and A. Hasan, *Security Analysis of XTR Exponentiation Algorithms against Simple Power Analysis Attack*, Preprint of CACR, Univ. of Waterloo, CACR 2004-05.
3. C. Clavier and M. Joye, *Univeral Exponentiation Algorithm A First Step towards Provable SPA-Resistance*, Cryptographic Hardware and Embedded Systems (CHES'01), LNCS2162, (2001), 300-308.

4. S. Chari, C.S. Jutla, J.R. Rao, and P. Rohatgi, *Towards sound approaches to counteract power-analysis attacks*, Advances in Cryptology - CRYPTO '99, LNCS1666, (1999), 398-412.
5. P.-A. Fouque and F. Valette, *The Doubling Attack Why Upwards is better than Downwards*, Workshop on Cryptographic Hardware and Embedded Systems 2003 (CHES 2003), LNCS 2779, (2003), 269-280.
6. G. Gaubatz, J.-P. Kaps, and B. Sunar, *Public Key Cryptography in Sensor Networks-Revisited*, 1st European Workshop on Security in Ad-Hoc and Sensor Networks, (ESAS 2004), LNCS3313, (2004), 2-18.
7. R. Granger, D. Page, and M. Stam, *A Comparison of CEILIDH and XTR*, Algorithmic Number Theory, (ANTS 2004), LNCS 3076, (2004), 235-249.
8. D.-G. Han, T. Izu, J. Lim, and K. Sakurai, *Side Channel Cryptanalysis on XTR Public Key Cryptosystem*, IEICE Trans. Fundamentals, Special Section on Discrete Mathematics and Its Applications, VOL.E88-A, NO.5, May, pp.1214-1223, (2005).
9. D.-G. Han, T. Takagi, T.H. Kim, H.W. Kim, and K.I. Chung, *Collision Attack on XTR and a Countermeasure with a Fixed Pattern*, International Association for Cryptologic Research (IACR), Cryptology ePrint Archive 2005/316, (2005). `http://eprint.iacr.org/2005/316`
10. Kocher, C., *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*, Advances in Cryptology - CRYPTO '96, LNCS 1109, (1996), 104-113.
11. Kocher, C., Jaffe, J., Jun, B., *Differential Power Analysis*, Advances in Cryptology - CRYPTO '99, LNCS1666, (1999), 388-397.
12. A.K. Lenstra and E.R. Verheul, *The XTR public key system*, Advances in Cryptology - CRYPTO '00, LNCS1880, (2000), 1-19.
13. K. Okeya and T. Takagi, *The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks*, CT-RSA 2003, LNCS 2612, (2003), 328-342, 2003.
14. D. Page and M. Stam, *On XTR and Side-Channel Analysis*, Pre-proceedings of SAC 2004, 67-81.
15. M. Stam and A.K. Lenstra, *Speeding Up XTR*, Proceedings of Asiacrypt 2001, LNCS2248, (2001), 125-143.
16. K. Schramm, T. Wollinger, and C. Paar, *A New Class of Collision Attacks and its Application to DES*, Proceedings of FSE 2003, LNCS2887, (2003), 206-222.
17. C.D. Walter, *Simple Power Analysis of Unified Code for ECC Double and Add*, Workshop on Cryptographic Hardware and Embedded Systems 2004 (CHES 2004), LNCS 3156, (2004), 191-204.
18. R. Watro, D. Kong, S-f. Cuti, C. Gardiner, C. Lynn, and P. Kruus, *TinyPK: Securing Sensor Networks with Public Key Technology*, ACM Workshop on Security of Ad Hoc and Sensor Networks 2004 (SASN 2004), 59-64.
19. S.-M. Yen, S. Kim, S. Lim, and S. Moon, *A Countermeasure against One Physical Cryptanalysis May Benefit Another Attack*, Information Security and Cryptology 2001 (ICISC 2001), LNCS 2288, (2001), 414-427.