An Equidistant Message Power Attack Using Restricted Number of Traces on Reduction Algorithm*

Jong-Yeon Park[1], Dong-Guk Han[1], Okyeon Yi[1], and Dooho Choi[2]

[1] Cryptography and Information Security Institute(CISI),
Department of Mathematics Kookmin University, Seoul, KOREA
{flyfree,christa,oyyi}@kookmin.ac.kr
[2] Electronic and Telecommunication Research Institute(ETRI).,
138 Gajeongno, Yuseong-gu, Daejeon, KOREA
dhchoi@etri.re.kr

Abstract. The RSA-CRT algorithm has been widely used for the efficiency of its exponent operation. Research has been announced about the physical susceptibility of RSA-CRT from various side channel attacks. Among them, Boer et al. proposed a brilliant differential power analysis (DPA) of CRT reduction with equidistant chosen messages that is called MRED (Modular reduction on Equidistant Data). This attack targets intermediate data that depend on the $[r = x \mathrm{mod} p]$ value. We introduce a new approach the MRED attack related to a subtraction algorithm which is not related only to the r value. This is superficially similar to previous DPA attacks but is based on a totally different assumption from the data dependent analysis. According to our result, only 256 traces are needed to reduce 1 block key to 2 key candidates, so it is a more efficient analysis method on restricted trace environments. Moreover, it is possible to attack a data dependent trace system. One example for this kind of attack is non-hamming weight. We describe our technique with its advantages and disadvantages, and show simulation results from a MSP430 software board.

Keywords: *Side channel attack(SCA), RSA-CRT, Differential Power Analysis(DPA), Correlation Power analysis(CPA), Modular Reduction on Equidiatant Data(MRED), Non-invasive attack*

1 Introduction

Side channel analysis (SCA) is not a traditional cryptanalysis technique, which would use only the inputs and outputs of cryptographic algorithms. SCA is in-

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0026354) This work was also supported by SCARF project which is the Research and Development program of KCC/KCA [Development of the Technology of Side Channel Attack Countermeasure Primitives and Security Validation]

stead a key searching technique that is based on the general characteristics or statistical analysis of a power signal and electromagnetic information from the operating cryptographic devices[6]. One of them, Differential power analysis(DPA) is among the strongest of these attacks. It is related to several methods such as Correlation power analysis (CPA)[7, 3]. In these attacks intermediate values are computed with guessed keys and analyzed power signal information. Several algorithms can be targeted in this attack and RSA, the most common public key cryptosystem, cannot avoid SCA. CRT based RSA algorithms are widely used due to its computing efficiency in several systems[8]. Because its intermediate data is hidden by the secret prime p, one cannot guess keys through an obvious DPA technique which is known as ZEMD[10]. Although a ZEMD attack cannot be applied, other threats exist from power analysis or electromagnetic analysis. The RSA-CRT algorithm must have two integral operations; reduction and recombination. These operations cause side channel leakages that can be exploited by techniques such as MRED(Modular reduction on Equidistant Data) power analysis by Boer et al[2], and Park el al. showed many ghost key patterns from MRED attack from the algorithms and selected bits[5]. Also, adaptive chosen plaintext attack by Novak[9] is suggested. The recombination step can also be attacked by multiplicative operations from Garner's CRT algorithm[4].
This paper suggests a modified MRED analysis that is not based on a data dependent signal, which we call Subtraction algorithm Analysis on Equidistant Data(SAED). SAED is focused on the subtraction algorithm located in the reduction algorithm, it is basically originated by Park et al[5]. It is different from a normal DPA attack using a data dependent attack model such as hamming weight or hamming distance. As a result of our experiments, we can reduce the secret key spaces using the even smaller number of traces than MRED.
The rest of this paper is organized as follows. In section 2, we describe and review the basic knowledge related to this paper. Section 3 explains our new attack method using a theoretical approach. In section 4, we show the experimental results of SAED analysis using a software ship board. Section 5 concludes this paper.

2 Prerequisites and Preliminaries

2.1 MRED analysis

Bore et al.[2] presented a brilliant attack technique on RSA-CRT[8]. This paper presents give you a short review of this method. It targets the initial reduction operation of CRT with the following steps. By inputting messages $x \mathrm{mod} p = r$ where $i \leq p$,

$$(x - i) \, mod \, p = (r - i) \tag{1}$$

One can guess the least significant byte of the value of r without the secret p value. By Equation 1, one can use the relation between inputs and reduction

outputs. Thus, the intermediate value set for r is $\{v_{i,j}\} = \{(j-i)\text{mod}256 | i = 0, ..., N-1, j = 0, ..., K-1\}$. Then, one computes the selected bit hamming weight for CPA. Table 1 shows how to compute intermediate values for, $v_{i,j}$. If the r is correctly guessed, the hamming weight of $v_{i,j}$ should be related to the power trace at the time of the implementation, which is the same principle as traditional DPA or CPA. Table 2 is the hamming weight set of 8-bit $v_{i,j}$,which is denoted by $h_{i,j}$.

Table 1. Table of $v_{i,j}$

$v_{i,j}$	x_0	x_1	x_2	x_3	x_4	$\cdots$	x_i
$v_{i,0}$	0	255	254	253	252	$\cdots$	$-i \mod 256$
$v_{i,1}$	1	0	255	254	253	$\cdots$	$(1-i) \mod 256$
$v_{i,2}$	2	1	0	255	254	$\cdots$	$(2-i) \mod 256$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$v_{i,255}$	255	254	253	252	251	$\cdots$	$(255-i) \mod 256$

Table 2. Table of the 8bit selection $h_{i,j}$ on the base of $v_{i,j}$

$h_{i,j}$	x_0	x_1	x_2	x_3	x_4	$\cdots$	x_i
$h_{i,0}$	0	8	7	7	6	$\cdots$	$HW(v_{i,0})$
$h_{i,1}$	1	0	8	7	7	$\cdots$	$HW(v_{i,1})$
$h_{i,2}$	1	1	0	8	7	$\cdots$	$HW(v_{i,2})$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$h_{i,255}$	8	7	7	6	7	$\cdots$	$HW(v_{i,255})$

For the second byte, the attack is repeated under the same conditions except for the distances of input values. It is intuitively clear that one can find extra bytes using the same technique. Equation 2 is a generalized form of Equation 1 that can be used to compute intermediate values for the attack.

$$x - i(256)^d \, mod \, p = r - i(256)^d \tag{2}$$

(In these tables $i(256)^d \leq p$ and d is the byte-index from the least significant byte. For example, if d is 0 the attack target is the least significant byte.) Because Equation 1 always works where $r \geq i(256)^d$ is valid, and $r \geq i(256)^d$ is valid when it is sufficiently large, one can guess the r value with a number of traces up to $r < N(256)^d$. Finding r gives the secret p directly by $GCD(x-r, N) = p$.

3 Subtraction algorithm Analysis on Equidistant Data(SAED)

3.1 Constant subtraction by equidistant message inputs

The reduction algorithm is used to compute the remainder which is a part of the division algorithm. Table 3 shows the multiple-precision division algorithm[1]. The reduction algorithm generally consists of multiple-precision multiplications, additions and subtractions. The last part of the reduction algorithm has subtraction by a constant value depending on the quotient (see Table 3 step 3.3). The following equidistant level is $r - 1 = x - 1 \bmod p$. We can see this is computed by the same quotient q, $x - 1 = qp + (r - 1)$. Although equidistant inputs are processed through many steps, q is not changed until the equidistant level is higher than p. Thus, we can regard step 3.3 $x \leftarrow x - q_{i-t-1}pb^{i-t-1}$ as just $x \leftarrow x - c$ because p is fixed. Focusing on the last iteration, r is finally produced by $x - c$, which we call $r = u - c$.

Table 3. multiple-precision division

INPUT : positive integers $x = (x_n \cdots x_1 x_0)_b, p = (p_t \cdots p_1 p_0)_b$ with $n \geq t \geq 1, p_t \neq 0$.
OUTPUT : the quotient $q = (q_{n-t} \cdots q_1 q_0)_b$ and remainder $r = (r_t \cdots r_1 r_0)_b$
such that $x = qp + r, 0 \leq r \leq p$.
1. For j from 0 to $(n - t)$ do : $q_j \leftarrow 0$
2. While $(x \geq pb^{n-t})$ do the following: $q_{n-t} \leftarrow +1, x \leftarrow x - pb^{n-t}$
3. For i from n down to $(t + 1)$ do the following:
3.1 If $x_i = p_t$ then set $q_{i-t-1} \leftarrow b - 1$; otherwise set $q_{i-t-1} \leftarrow \lfloor (x_i b + x_{i-1}/p_t) \rfloor$.
3.2 While $(q_{i-t-1}(p_t b + p_{t-1}) > x_t b^2 + x_{i-1} b + x_{i-2})$ do: $q_{i-t-1} \leftarrow q_{i-t-1} - 1$.
3.3 $x \leftarrow x - q_{i-t-1}pb^{i-t-1}$.
3.4 If $x < 0$ then set $x \leftarrow x_p b^{i-t-1}$ and $q_{i-t-q} \leftarrow q_{i-t-1} - 1$.
4. $r \leftarrow x$
5. Return (q, r)

3.2 Basic Principal of SAED

In a general DPA attack (including MRED), one has to compute the hamming weight of intermediate data. However, our approach does not need to consider hamming weight and the data dependent power signal. Therefore one need only use Table 1 instead of Table 2 in section 2.1.

Table 4 is a subtraction algorithm. This algorithm is a part of the reduction, especially in step 3.3 in Table 3. In step 2.2, starting from the most significant

Table 4. multiple-precision subtraction

INPUT : positive integers u and c, each having $n+1$ base b digits, with $u \geq c$.
OUTPUT : the difference $u - c = (r_1 r_{n-1} ... r_{n-1})_b$ in radix b representation.
1. $BR \leftarrow 0$.
2. For i from 0 to n do the following:
2.1 $r_i \leftarrow (u_i - c_i + b) \mathrm{mod} b$.
2.2 If $(u_i - c_i + b) \geq 0$ then $BR \leftarrow 0$; otherwise $BR \leftarrow -1$.
3. Return$(r_1 r_{n-1} ... r_{n-1})_b$

byte, the algorithm selects borrow, presented 'BR', between 0 or -1. We assume that the power signal is distinguished by borrow determination in step 2.2. We focus only on a one byte operation, $u_0 - c_0$. If c_0 is constant, the borrow determination is influenced by u_0. Therefore we can only know the signal turn if- to otherwise-, it only occurs where $u_0 < c_0$.

Table 5. Borrow occurrence where $(u_0 - c_0) \mathrm{mod} 256$

u_0	255	254	253	$\cdots$	120	119	118	$\cdots$	0	255	254	$\cdots$
c_0	120	120	120	$\cdots$	120	120	120	$\cdots$	120	120	120	$\cdots$
$BorrowOccurence$	N	N	N	$\cdots$	N	Y	Y	$\cdots$	Y	N	N	$\cdots$

Table 5 shows an example where u_0 is $255-i$ and there is a borrow occurrence with respect to the operation, $u_0 - c_0 \mathrm{mod} 256$, $c_0 = 120$. The u_0 sequence by equidistant message attack is $\{255 - i\}$ and c_0 is fixed by the quotient q of the reduction algorithm in reduction $r = x \mathrm{mod} p$ that is expressed by $x = p \times q + r$. Equidistant inputs mediate the u_0 series $\{u_0$,$u_0 - 1\mathrm{mod}256$,$u_0 - 2\mathrm{mod}256$,$u_0 - 3\mathrm{mod}256$,$\cdots\}$ and c_0 is fixed by the quotient. Because borrow occurrence is determined by the u_0 series varying, power traces are distinguished by [The Event] being Yes or No. Therefore, if the attacker can guess classify the trace into Yes or No, a general CPA attack is available. However,c_0 is unknown because one cannot know the divisor p, so the borrow occurrence is unpredictable. Now the exact c_0 is not possible to know, but one applies stochastic computation alternately. The probability of the borrow occurrence of $y - c_0$ is equation3 whith the arbitrary value u_0 in Z_{256} and c_0 .

$$P(BO) = P(u_0 < c_0) = (255 - u_0)/256 \tag{3}$$

The picture of the probability of cases of Borrow occurrence by c_0 and fixed u_0 is shown in Figure 1, provided that $c_0 \in Z_{256}$ is following uniform distribution.

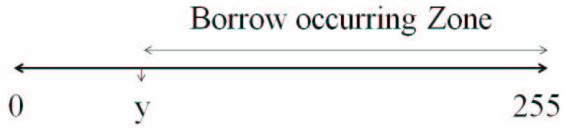

Fig. 1. Cases of Borrow occurrence by uniformly distributed c_0 and fixed y

The least significant byte of q_0p is c_0 , and the multiplication of 2 unknown values can be seen in Table 3. Actually, LSB of multiplication of 2 random values is not a perfect uniform distribution. Although its real distribution is not uniform, in the interval, (for example, 0 to 50, 50 to 100, 100 to 200, 200 to 255) it behaves as a uniform distribution. Moreover, the output result is somewhat uniform distributed from the 2nd byte. Therefore we assume each blocks of c are uniformly distributed on each result.
Using the characteristics, although c_0 is known intermediate values can still be computed with a stochastic tool.

$$SU_{255} = \{0/256, 1/256, 2/256, \cdots, 254/256, 255/256, 0/256, \cdots\} \tag{4}$$

SU_{255} is a sequence by stochastic computation for the maximum correlation coefficient between borrow occurring trace information and intermediate data by equation 3. In this way, SU_j is a new metric to find u_0 . However, one is not interested in the u_0 value, because r_0 is still not computable since c_0 is totally unknown. Therefore, we pay attention to the relation between u_0 and r_0 .

Theorem 1 (Coincidence property of Borrow and Carry). *In multiply-precision operation algorithms of multiply-precision data a,b and c, borrow occurs in a block operation of big numbers $a - c = b$ if and only if carry occurs the same equation $a = b + c$ on a correspondent block addition.*

proof)Suppose that borrow occurs on a byte of $a - c = b$, denote $a_0 - c_0 = b_0 \text{mod} 256$. By using $a_0 < c_0$, $a_0 = c_0 + b_0 - 256$ is naturally induced. This means a carry occurrence of a = b + c in the correspondence block operation. ■

Theorem 1 is a good bridge between the borrow occurrence of subtraction and the carry occurrence of addition. The power signal has only borrow occurrence information but we can regard this as the carry of the addition algorithm. The probability of a carry occurrence, of $r_0 + c_0$ is equation5 with an arbitrary value r_0 in Z_{256} and c_0.

$$P(CO) = P(256 \leq r_0 + c_0) = (256 - r_0 \leq c_0) = r_0/256 \tag{5}$$

for example, $u_0 = 255$, $c_0 = 120$ and $u_0 - c_0 = r_0 = 135$. To guess the key(r_0), the intermediate value of the equidistant sequence follows from equation 6.

$$SR_{135} = \{135/256, 134/256, 133/256, \cdots, 1/256, 0/256, 255/256, \cdots\} \quad (6)$$

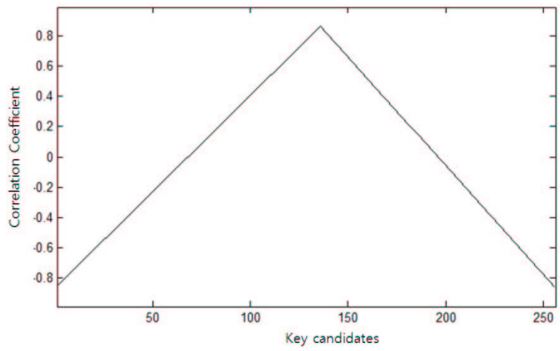

Fig. 2. Correlation coefficient of SR_j with borrow occurrence

Finally, we get the sequence $SR_j, (0 \leq j \leq 255)$, from this one can find the key with the correlation coefficient between the borrow occurring trace and SR_j. Figure 2 is the simulation correlation coefficient result set by $u_0 = 255$, $c_0 = 120$ and $u_0 - c_0 = r_0 = 135$. The result is the highest peak is 135 and that is the key we would like to get. On the other hand, the lowest peak is 255 and that is u_0. This is because $\rho(SU_j, SR_j) = -1$, so $\rho(SR_{135}, BorrowOccurence) = -\rho(SU_{255}, BorrowOccurence)$.

4 Experimental result and limitations

From section 3.2, one can construct intermediate data from the subtraction algorithm and theoretical approach of SAED. In this section, we describe the real experimental result and analyze the performance and efficiency of the attack.

Table 6. Experimental Environment

Signal Acquisition	Digital oscilloscope Lecroy
Board	MCU chip board(MSP 430 - software board)
Sampling rate	250MS/sec
Algorithm	8bit modular reduction algorithm by equidistant chosen message, $x - i(256)^k \mathrm{mod} p = r - i(256)^k$
Size of the variables	32byte equidistant inputs 16byte prime p

Table 6 shows the experimental setup. This experiment is performed by a modular reduction algorithm on an MCU chip board. The result of our attack was to find , 2 final keys by comparing the maximum correlation coefficient to the subtraction time period of the reduction algorithm, shown figure 3. As well as this, high correlations are distributed around the right key candidates. These can act as clues to determine just one exact key. Two maximum points on figure 3 are $u_0(255)$ and $r_0(135)$, which must be symmetric theoretically.

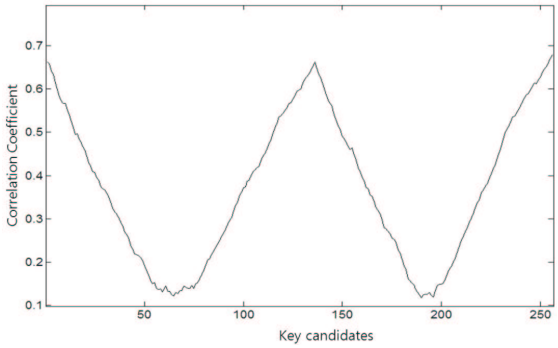

Fig. 3. Maximum correlation of 256 key candidates by SAED

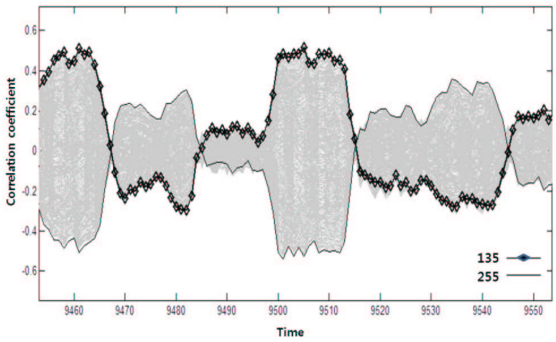

Fig. 4. Correlation coefficient of 256 key candidates on time domain

Figure 4 shows the correlation coefficient against the time domain of 256 key candidates simultaneously. The two plotted lines are the correlation coefficient computed at SR_{255} and SR_{135}, the other keys are distributed in gray zone. This symmetrical characteristic removes 2 key candidates from the 256 keys. One of them must be r_0 and the other one is u_0. This is a weakness of our method because MRED gives only one right key. Moreover, if c_0 is 0, there is

Table 7. Minimum required traces comparison between SAED and MRED

	$r_0 = 135, u_0 = 255, c_0 = 120$	$r_1 = 2, u_2 = 98, c_1 = 96$	$r_2 = 105, u_2 = 176, c_2 = 71$
SAED	256	256	256
MRED	over 2800	over 3000	over 1800

no intermediate difference data from SAED because borrow does not occur in the subtraction algorithm.

SAED has 2 major advantages however. First, its totally different assumption which does not follow the hamming weight model. Second, the results analysis shows outstanding efficiency, as seen Table 7. In MRED, one needs thousands of traces to find a 1 byte key. On the other hand, only 256 traces are needed in SAED for the same attack target. As a result, $256 \times n$(Number of bytes blocks)traces are need to find the complete r. This is a dramatic enhancement in the required traces efficiency, where MRED needs $\times 10$ more traces compared with SAED. If c_0 is biased to 255 or 0, SAED's performance could decrease however.

5 Conclusion

This paper suggests a new equidistant message attack which uses new assumptions and modified measures while using existing equidistant chosen message attack techniques. From our result, only $256 \times n$ traces are needed for finding 2^n key candidates. Moreover, SAED has comparatively long term peaks, which is an advantage for key searching compared with MRED. However, SAED has disadvantages, such as key candidate problems and efficiency depending on the constant value. Therefore, we compensate for these defects, by studying compensated methods derived from SAED.

References

1. A.J Menezes, Paul C.van Oorschot and S.A Vanstone, *Handbook Applied Cryptography*, CRC press ISBN: 0-8493-8523-7, 1996.
2. B. D. Boer, K. Lemke, and G. Wicke , *A DPA attack against the modular reduction within a crt implementation implementation of RSA* , Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS2523, 228-243, 2002.
3. E. Brier, C. Clavier, and F. Olivier, *Correlation power analysis with a leakage model* , Workshop on Cryptographic Hardware and Embedded Systems(CHES 2004), LNCS3156, 16-29, 2004.
4. F. Amiel, B. Feix, and K. Villegas, *Power Analysis for Secret Recovering and Reverse Engineering of Public Key Algorithms* , Selected Area Cryptography(SAC 2007), LNCS4876, 110-125, 2007.

5. J. Park, D. Han, O. Yi, D Choi, *Ghost Key Patterns of MRED Power Analysis on RSA-CRT*, Symposium on Cryptography and Information Security(SCIS 2011), Kitakyushu Japan, 2011.
6. P,Kocher, J.Jaffe, and B. Jun, *Timming attacks on implementations of diffe-hellman, rsa, dss, and other systems*, CRYPTO 96 Proceedings of the 16th Annual International cryptology Conference on Advances in Cryptology. ISBN 3-540-61512-1.
7. P,Kocher, J.Jaffe, and B. Jun, *Introduction to differential power analysis and related attacks* , 1998, White paper, Cryptography Research, http://www.cryptography.com/dpa/ technical, 1998.
8. Rivest R, Shamir A, Adleman L, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM vol 21, Issue 2, 120-126, 1978.
9. R. Novak, *SPA-Based Adaptive Chosen- Ciphertext Attack on RSA Implementation*, Public Key Cryptography (PKC 2002), LNCS2274, 252-262, 2002.
10. T.S.Messerges, E.A. Dabbish and R.H. Sloan, *Power Analysis Attacks of Modular Exponentiation in Smartcards* , Workshop on Cryptographic Hardware and Embedded Systems (CHES 99), LNCS1717, 144-157, 1999.