A DPA Attack on the Improved Ha-Moon Algorithm*

Dong Jin PARK and Pil Joong LEE

Information Security Laboratory,
Dept. of EEE, POSTECH, Pohang, Korea
djpark@oberon.postech.ac.kr, pjl@postech.ac.kr

Abstract. The algorithm proposed by Ha and Moon [2] is a countermeasure against power analysis. The Ha-Moon algorithm has two drawbacks in that it requires an inversion and has a right-to-left approach. Recently, Yen, Chen, Moon and Ha improved the algorithm by removing these drawbacks [7]. Their new algorithm is inversion-free, has a left-to-right approach and employs a window method. They insisted that their algorithm leads to a more secure countermeasure in computing modular exponentiation against side-channel attacks. This algorithm, however, still has a similar weakness observed in [1, 6]. This paper shows that the improved Ha-Moon algorithm is vulnerable to differential power analysis even if we employ their method in selecting s_i.

Keywords: Ha-Moon algorithm, randomized exponentiation algorithm, side-channel attack.

1 Introduction

In 2002, Ha and Moon proposed an algorithm in order to prevent power analysis [2]. The Ha-Moon algorithm randomizes a secret exponent into a signed binary representation. Many researchers are interested in this algorithm because of its simplicity and efficiency. Two drawbacks of the Ha-Moon algorithm are that it requires an inversion of a group element and recodes an exponent into a randomized representation from LSB to MSB (i.e. right-to-left).

Recently, Yen, Chen, Moon and Ha improved these drawbacks of the Ha-Moon algorithm [7]; their new algorithm (improved Ha-Moon algorithm) has a left-to-right approach and does not require an inversion of a group element. Thus, their algorithm can be applied in computing modular exponentiations, such as RSA and DSA. They insisted that their algorithm leads to a more secure countermeasure implementing exponentiation against side-channel attacks. However, this paper shows that the improved Ha-Moon algorithm is still vulnerable to differential power analysis (DPA) [3, 4]. Thus, the improved Ha-Moon algorithm should not be implemented in restricted environments, such as smart cards, for which it was originally designed.

---

* This research was supported by University IT Research Center Project, the Brain Korea 21 Project.

The remainder of this paper organized as follows: In Section 2, we briefly review the improved Ha-Moon algorithm. In Section 3, we propose an attack method that shows the improved Ha-Moon algorithm is still vulnerable to DPA.

2 Improved Ha-Moon Algorithm

This section summarizes the improved Ha-Moon Algorithm. See [7] for details.

2.1 Brief Description

Algorithm 1. Improved Ha-Moon algorithm with 2-bit window (Fig. 3 in [7])

INPUT: $g, K = (k_{n-1}, \cdots, k_0)_2$ where n is even and $(k_{n-1}k_{n-2})_2 = (01)_2, (10)_2$, or $(11)_2$
OUTPUT: g^K

1. $R[0] = 1; R[1] = g$
2. Precomputation: $R[2] = g^2, \cdots, R[14] = g^{14}$
3. $s = -(k_{n-1}k_{n-2})_2$
4. for i from $n-4$ downto 0 step -2 do
   4.1 $d = -4s$
   4.2 $s =$RandomInteger$(-1, -3)$
   4.3 $R[0] = R[0]^4$
   4.4 $R[0] = R[0] \times R[d + s + (k_{i+1}k_i)_2]$
5. $R[0] = R[0] \times R[-s]$
6. output $R[0]$

The improved Ha-Moon algorithm is a left-to-right, inversion-free, and window[1] method. In this algorithm, a randomized exponent d_i' is recoded from the following equation:

$$d_i' - s_i = (k_{i+1}k_i)_2 - 4s_{i+2}$$

where $(k_{i+1}k_i)_2$ is a secret exponent to be recoded and $s_i \in_R \{-1, -2, -3\}$ which introduces randomness in the representation. Since d_i' becomes a positive integer for all i, there is no inversion operation in Algorithm 1. In Algorithm 1, there are always two squarings and multiplication sequences, which are not dummy operations. Thus, the improved algorithm can resist SPA-like attacks, such as [5], and the safe-error attack [8]. Also, the improved Ha-Moon algorithm may resist Fouque *et al.*' attack [1], because each probability of state transitions seems to be equal.

[1] We assume without loss of generality that the window size is 2.

2.2 Weakness of the Improved Ha-Moon Algorithm

However, the improved Ha-Moon algorithm has a weakness similar to the original Ha-Moon algorithm in that there are a few possible intermediate values [1, 6]. After processing $(k_{i+1}k_i)_2$ in Step 4.4, Algorithm 1, $R[0]$ becomes one of $g^{(k_{n-1}\cdots k_i)_2-1}$, $g^{(k_{n-1}\cdots k_i)_2-2}$, and $g^{(k_{n-1}\cdots k_i)_2-3}$. In other words, there are only three possible intermediate values in any iteration. Table 1 shows different pattern of intermediate values according to $(k_{i+1}k_i)_2$. Each occurrence of $g^{4(k_{n-1}\cdots k_{i+2})_2+x_i}$ given $(k_{n-1}\cdots k_{i+2})_2$ can be checked by DPA, such as ZEMD attack [4]. For example, $(k_{i+1}k_i)_2 = 0$ results peaks in $x_i = -3$, -2, and -1 and $(k_{i+1}k_i)_2 = 1$ in $x_i = -2$, -1, and 0. Thus, we can find a correct $(k_{i+1}k_i)_2$ given $(k_{n-1}\cdots k_{i+2})_2$.

Note that, in this attack, a third of the samples are meaningful and the others are treated as noise, because the possible distribution of intermediate values is three.

Table 1. Intermediate values, $g^{4(k_{n-1}\cdots k_{i+2})_2+x_i}$, after processing $(k_{i+1}k_i)_2$

	x_i					
$(k_{i+1}k_i)_2$	-3	-2	-1	0	1	2
0	○	○	○			
1		○	○	○		
2			○	○	○	
3				○	○	○

2.3 Yen *et al.*'s Method

Yen *et al.* suggested a method to prevent this attack. Their method is selecting $s_i = -1$ or -2 when $(k_{i+1}k_i)_2 = 0$ or 2 as well as selecting $s_i = -2$ or -3 when $(k_{i+1}k_i)_2 = 1$ or 3. The allowed parameters are summarized in Table 2. Their method can make $(k_{i+1}k_i)_2 = 0$ and 1(2 and 3) indistinguishable. For this reason, they insisted that the attack in the previous section can be avoided by this method.

3 Proposed Attack

Unfortunately, Yen *et al.*'s method does not provide additional randomness in the intermediate values. The indistinguishability after processing $(k_{i+1}k_i)_2$ can be removed in the successive iteration. After processing $(k_{i-1}k_{i-2})_2$ in Step 4.4, Algorithm 1, $R[0]$ becomes

$$g^{16(k_{n-1}\cdots k_{i+2})_2+4(k_{i+1}k_i)_2+(k_{i-1}k_{i-2})_2+s_{i-2}}$$

Table 2. Parameters with the Yen *et al.*'s method

s_{i+2}	$(k_{i+1}k_i)_2$	(s_i, d'_i)
-1	0	$(-2, 2)$ or $(-1, 3)$
-1	1	$(-3, 2)$ or $(-2, 3)$
-1	2	$(-2, 4)$ or $(-1, 5)$
-1	3	$(-3, 4)$ or $(-2, 5)$
-2	0	$(-2, 6)$ or $(-1, 7)$
-2	1	$(-3, 6)$ or $(-2, 7)$
-2	2	$(-2, 8)$ or $(-1, 9)$
-2	3	$(-3, 8)$ or $(-2, 9)$
-3	0	$(-2, 10)$ or $(-1, 11)$
-3	1	$(-3, 10)$ or $(-2, 11)$
-3	2	$(-2, 12)$ or $(-1, 13)$
-3	3	$(-3, 12)$ or $(-2, 13)$

where $s_{i-2} \in \{-1, -2, -3\}$. Table 3 shows possible values of $R[0]$ after processing $(k_{i-1}k_{i-2})_2$. If $(k_{n-1} \cdots k_{i+2})_2$ is known, we can determine $(k_{i+1}k_i)_2$ and classify $(k_{i-1}k_{i-2})_2$ into a group A (0 or 1) or a group B (2 or 3).

Algorithm 2. ZEMD-like attack on the improved Ha-Moon algorithm

OUTPUT: K

1. gather sufficiently many power trace samples of g_w^K for different g_w's.
2. for i from $n-2$ downto 2 step -2 do
   - 2.1 for x from -2 to 13 step 1 do
     - 2.1.1 divide the samples into two sets $S1$ and $S2$ according to a decision function, such as the Hamming weight of $g_w^{16(k_{n-1}\cdots k_{i+2})_2+x}$
     - 2.1.2 get the bias signal as $D = \text{average}(S1) - \text{average}(S2)$
     - 2.1.3 record an appearance of a spike in D
   - 2.2 determine $(k_{i+1}k_i)_2$ and classify $(k_{i-1}k_{i-2})_2$ into a group A or B according to records in Step 2.1.3
3. guess $(k_1k_0)_2$
4. output K

For example, if a spike is recorded in Step 2.1.3, Algorithm 2 when $x = 6$ and (or) 7, then we can find that $(k_{i+1}k_i)_2$ is 2 and $(k_{i-1}k_{i-2})_2$ is classified into a group A. Thus, we can determine a secret exponent K except $(k_1k_0)_2$, of which we can classify the group, A or B; the size of the search space from the remaining ambiguity in $(k_1k_0)_2$ is only two. In addition, our attack does not assume anything beyond ZEMD attack.

That is, the improved Ha-Moon algorithm is vulnerable to DPA. Yen *et al.*'s method does not prevent DPA. Rather it helps DPA to break the improved Ha-Moon algorithm by increasing the rate of meaningful power traces from a third to a half, because their method makes the possible distribution of intermediate values be two. Even enlarging the range of the intermediate values will not

Table 3. Intermediate values, $g^{16(k_{n-1}\cdots k_{i+2})_2+x_{i-2}}$, after processing $(k_{i-1}k_{i-2})_2$

	x_{i-2}															
$(k_{i+1}k_i)_2$	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	A	A	B	B												
1					A	A	B	B								
2									A	A	B	B				
3													A	A	B	B

A means $(k_{i-1}k_{i-2})_2 = 0$ or 1, and
B means $(k_{i-1}k_{i-2})_2 = 2$ or 3.

increase the complexity of DPA significantly, but only decrease the rate in inverse proportion to the range.

4 Conclusion

The improved Ha-Moon algorithm introduced interesting properties, such as a left-to-right approach, inversion-free and window method. This paper, however, shows that the improved Ha-Moon algorithm does not resolve one critical property of the Ha-Moon algorithm: its vulnerability to DPA. The improved Ha-Moon algorithm should be used with another randomizing countermeasure.

References

1. P.-A. Fouque, F. Muller, G. Poupard and F. Valette, "Defeating countermeasures based on randomized BSD representation," *CHES 2004,* LNCS 3156, pp. 312–327, Springer-Verlag, 2004.
2. J. C. Ha and S. J. Moon, "Randomized signed-scalar multiplication of ECC to resist power attacks," *CHES 2002,* LNCS 2523, pp. 551–563, Springer-Verlag, 2002.
3. P. Kocher, J. Jaffe and B. Jun, "Differential power analysis," *CRYPTO 1999,* LNCS 1666, pp. 388–397, Springer-Verlag, 1999.
4. T. S. Messerges, E. A. Dabbish and R. H. Sloan, "Power analysis attacks of modular exponentiation in smartcards," *CHES 1999,* LNCS 1717, pp. 144–157, Springer-Verlag, 1999.
5. K. Okeya and D.-G. Han, "Side channel attack on Ha-Moon's countermeasure of randomized signed scalar multiplication," *INDOCRYPT 2003,* LNCS 2904, pp. 334–348, Springer-Verlag, 2003.
6. S. G. Sim, D. J. Park and P. J. Lee, "New power analyses on the Ha-Moon algorithm and the MIST algorithm," *ICICS 2004*, LNCS 3269, pp. 291–304, Springer-Verlag, 2004.
7. S.-M. Yen, C.-N. Chen, S. Moon and J. Ha, "Improvement on Ha-Moon randomized exponentiation algorithm," *ICISC 2004,* to appear in LNCS, Springer-Verlag, 2004.
8. S.-M. Yen, S. Kim, S. Lim and S. Moon, "A countermeasure against one physical cryptanalysis may benefit another attack," *ICISC 2001,* LNCS 2288, pp. 414–427, Springer-Verlag, 2004.