New Single-Trace Side-channel Attacks on a Specific Class Elgamal

N. Mahdion, Hadi Soleimany, Pouya Habibi, Farokhlagha Moazami

Cyberspace Research Institute, Shahid Beheshti University, Tehran, Iran

p.mahdion@sbu.ac.ir, h_soleimany@sbu.ac.ir, f_moazemi@sbu.ac.ir

Shahid Beheshti University
Protein Research Center

1 Introduction

The main purpose of SCAs on RSA and Elgamal public key encryptions is modular exponentiation algorithms that involve multiplication and squaring operations. However most implantation of multiplication and square algorithm use the similar sequence instructions, it is very difficult to distinguish between multiplication and square process for random input messages[7]. In order to overcome this problem many researchers proposed some attacks such that distinguishing between square and multiply is depend on the chosen input message[8]. SPA with adaptively chosen messages are proposed by Novak[8], applicable on RSA implementation. There are some chosen message attacks on public key encryption in [4, 3, 11, 5]. In 2005, Yen et al. proposed the first $N-1$ attack on the modular exponentiation algorithms where N is the modulus in RSA or Elgamal and all its powers are either 1 or -1 [13]. The $N-1$ as input makes collisions in internal state of modular exponential which make it possible to distinguish between squaring and multiplying operations in only single-trace. Table 1 shows brief of this attack on SMA algorithm. Fig. 1 is an example for The $N-1$ attack against SMA algorithm in one power trace.

Table 1: Summary of$N-1$ attack on SMA Algorithm

d_i	S_{i-1}
$d_i=0$	$1^2 \mod n$
$d_i=1$	$(n-1)^2 \mod n$

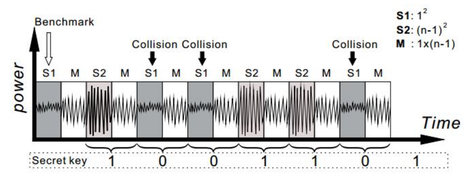


Figure 1: The $N-1$ attack on SMA algorithm in single power trace[2].

2 Main Objectives

Several researches presented many different countermeasures to guard the implementation of exponentiation functions against the $N-1$ attack. However, a strong countermeasure cannot be established for free. It has an impact on the latency, power consumption and size of the implementation. Consequently, simple countermeasures in real-world applications are preferred as they utilize a small overhead, rather than complicated solutions with notable extra cost. In this direction, to protect the implementation against the $N-1$ attack, several articles propose the simplest solution, i.e. "block the special message $N-1$".
This research demonstrates that SMA and ML algorithms are vulnerable by $N-1$-type attacks even if the special message $N-1$ is blocked. We illustrate the practical feasibility of our attacks in real_life experiments using SPA attack.

3 Mathematical Section

3.1 Our chosen ciphertext

Basic idea of our attack is similar to the original $N-1$ attack. We select a ciphertext that enhances the differences between executed operations during the modular exponentiation algorithms according to the bit pattern of the secret key. We utilize a chosen message c such that $c^2=-1 \mod p$ where p is $4k+1$. In the following, we show how create our ciphertext by using Little Fermat Theorem.

Little Fermat Theorem: If p is a prime number and a is any number not divisible by p, then:

$$a^{(p-1)} \mod p = 1$$

Based on the Little Fermat Theorem, we can conclude $a^{\frac{p-1}{2}} = \pm 1 \bmod p$. In mathematics, if $a^{\frac{p-1}{2}} = 1 \bmod p$, a is called a quadratic residue $\bmod p$ and if $a^{\frac{p-1}{2}} = -1 \bmod p$, a is called a non-residue $\bmod p$. Let a be a non-residue $\bmod p$ so $a^{\frac{p-1}{2}} = p-1 = -1 \bmod p$, when $p=4k+1$, we have:

$$a^{\frac{p-1}{2}} = a^{\frac{4k+1-1}{2}} = a^{\frac{4k}{2}} = a^{2k} = (a^k)^2 \bmod p.$$

Let $a^k = c$, therefore we have $c^2 = -1 = p-1 \bmod p$ and c is the appropriate number to choose as a distinguisher.

3.2 Attack on Square and Multiply Always Algorithm

One of the efficient algorithms for modular exponentiation is the Square and Multiply Always algorithm (SMA). Algorithm 1 illustrates an implementation of the SMA algorithm.

Algorithm 1 : Square and Multiply Always (SMA)

Require: c, $d=(d_{l-1}...d_0)_2$, p .
Ensure: $m = c^d \bmod p$.
1: $m \leftarrow 1$
2: **for** $i = l-1$ downto 0 **do**
3: $\quad m \leftarrow m^2 \bmod p \quad \triangleright S_i$
4: $\quad t \leftarrow m \times c \bmod p \quad \triangleright M_i$
5: $\quad$ **if** $d_i = 1$ **then**
6: $\quad\quad m \leftarrow t$
7: $\quad$ **end if**
8: **end for**
9: **return** m

By crafting the ciphertext c such that $c^2 = p-1 \bmod p$, the output of squaring (line 3 of Algorithm 1) is always 1 or $p-1$. These values inter as input of the multiplication operation (line 4). Hence, when the bit of the secret key is 1 ($d_i=1$) then the input of the squaring operation is either c or $p-c$ and when the bit of the secret key is 0 ($d_i=0$) then the the input of squaring operation is either 1 or $p-1$.

Table 2: Summery of our attack on SMA Algorithm

d_i	M_{i-1}
0	$(1 \times c) \mod p$
1	$(p-1 \times c) \mod p$

Since $(p-1)$ is a large random-looking number in comparison to the value of 1, we expect that the differences in the patterns of two multiply operations $((p-1) \times c) \bmod p$ and $(1 \times c) \bmod p$. These differences can be easily perceived in a single-trace of side-channel information by visual observation.

3.3 Attack on Montgomery Ladder Algorithm

Another popular algorithm for modular exponentiation is the Montgomery Ladder algorithm (ML) which is demonstrated in Algorithm 2.

Algorithm 2 Montgomery Ladder (ML)

Require: c, p, $d=(d_l d_{l-1}...d_0)_2$ $(d_l=1)$.
Ensure: $m = c^d \bmod p$.
1: $R_0 \leftarrow c$
2: $R_1 \leftarrow R_0 \times R_0 \bmod p$
3: **for** $i = l-1$ downto 0 **do**
4: $\quad R_{1-d_i} \leftarrow R_0 \times R_1 \bmod p \quad \triangleright M_i$
5: $\quad R_{d_i} \leftarrow R_{d_i} \times R_{d_i} \bmod p \quad \triangleright S_i$
6: **end for**
7: **return** R_0

Let us assume that the ciphertext c such that $c^2 = -1 \bmod p$ is given as an input to the ML algorithm. By categorizing the executed operations for all of the modes two consecutive secret exponent bits d_i and d_{i-1}, we find out that the modular squaring operation can be exploited solely to reveal the secret exponent bits, as Table 3. In this Table, S_{i-1} denote squaring operation at the $(i-1)$-th iteration.

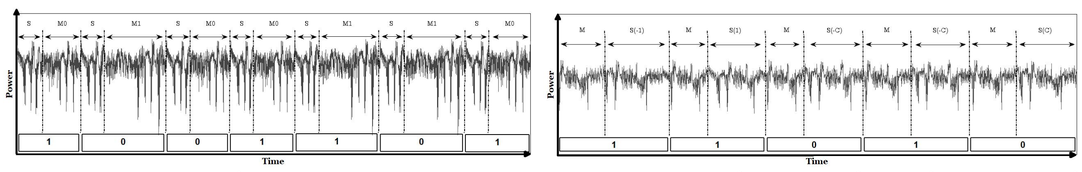
Table 3: Summery of our attack on ML Algorithm

$d_i \longrightarrow d_{i-1}$	S_{i-1}
$d_i = d_{i-1}$	(1×1) and $(p-1 \times p-1) \mod p$
$d_i \neq d_{i-1}$	$(c \times c)$ and $(p-c \times p-c) \mod p$

The operations in Table 3 can easily perceived in a single-trace of side-channel information by visual observation. Notice, our attack is also applicable on Montgomery Powering Ladder algorithm (Modified ML algorithms).

4 Experimental results

We verified our theoretical model by implementing SMA and ML algorithms on an Atmel ATXMEGA128D4 8-bit micro-controller which was located on the TARGET Board of the ChipWhisperer CW1173 [9]. Fig. 2 illustrates our proposed attacks on SMA and ML algorithms.



(a) Attack on SMA. S=1^2, M0=$(1 \times c)$ and M1= $(-1 \times c)$. **(b)** Attack on ML. S(1)=1^2, S(-1)=$(p-1)^2$, S(-1)=c^2 and S(-c)=$(p-c)^2$.

Figure 2: Experimental results of our proposed attacks

5 Conclusions

In this research, we proposed a new chosen ciphertext attack on Elgamal encryption which implements by using SMA and ML algorithms. Our ciphertext is c such that $c^2 = p-1 \mod p$, where p is the prime module and the public key in Elgamal cryptosystem. We exploited the leakage of power consumption to confirm the practicability of the proposed attack during the decryption execution of a specific ciphertext c.

References

[1] Jean-Christophe Courrège, Benoit Feix, and Mylène Roussellet. Simple power analysis on exponentiation revisited. In *CARDIS 2010*, pages 65–79, 2010.

[2] Zhaojing Ding, Wei Guo, Liangjian Su, Jizeng Wei, and Haihua Gu. Further research on N-1 attack against exponentiation algorithms. In *ACISP 2014*, pages 162–175, 2014.

[3] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation. In *CHES 2015*, pages 207–228, 2015.

[4] Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer. ECDH key-extraction via low-bandwidth electromagnetic attacks on pcs. In *CT-RSA 2016*, pages 219–235, 2016.

[5] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA key extraction via low-bandwidth acoustic cryptanalysis. *IACR Cryptology ePrint Archive*, 2013:857, 2013.

[6] Naofumi Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, and Adi Shamir. Collision-based power analysis of modular exponentiation using chosen-message pairs. In *CHES 2008*, pages 15–29, 2008.

[7] Naofumi Homma, Atsushi Miyamoto, Takafumi Aoki, Akashi Satoh, and Adi Shamir. Comparative power analysis of modular exponentiation algorithms. *IEEE Trans. Computers*, 59(6):795–807, 2010.

[8] Roman Novak. Spa-based adaptive chosen-ciphertext attack on RSA implementation. In *PKC 2002*, pages 252–262, 2002.

[9] Colin O'Flynn and Zhizhang (David) Chen. Chipwhisperer: An opensource platform for hardware embedded security research. In *COSADE 2015*, pages 243–260, 2015.

[10] Werner Schindler. A timing attack against RSA with the chinese remainder theorem. In *CHES 2000*, pages 109–124, 2000.

[11] Werner Schindler. Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. In *CHES 2015*, pages 229–247, 2015.

[12] Sung-Ming Yen, Lee-Chun Ko, Sang-Jae Moon, and JaeCheol Ha. Relative doubling attack against montgomery ladder. In *ICISC 2005*, pages 117–128, 2005.

[13] Sung-Ming Yen, Wei-Chih Lien, Sang-Jae Moon, and JaeCheol Ha. Power analysis by exploiting chosen message and internal collisions - vulnerability of checking mechanism for rsa-decryption. In *Mycrypt 2005*, pages 183–195, 2005.