Amplitude Demodulation-based EM Analysis of different RSA implementations

Guilherme Perin, Lionel Torres, Pascal Benoit and Philippe Maurine
LIRMM, University of Montpellier 2
161, Rue Ada 34095, Montpellier, France
Email: perin, lionel.torres, pascal.benoit, philippe.maurine@lirmm.fr

Abstract—**This paper presents a fully numeric amplitude-demodulation based technique to enhance simple electromagnetic analyses. The technique, thanks to the removal of the clock harmonics and some noise sources, allows efficiently disclosing the leaking information. It has been applied to three different modular exponentiation algorithms, mapped onto the same multiplexed architecture. The latter is able to perform the exponentiation with successive modular multiplications using the Montgomery method. Experimental results demonstrate the efficiency of the applied demodulation based technique and also point out the remaining weaknesses of the considered architecture to retrieve secret keys.**

Keywords: Public-Key Cryptography, RSA, Modular Exponentiation, Side-Channel Attacks, AM Demodulation.

I. Introduction

Side-channel Analysis (SCA) has been widely adopted to attack cryptographic systems. Although cryptographic devices leak information through different channels, power analyses (SPA, DPA [1], CPA [2]) were recognized as the most efficient ones. However, Electromagnetic (EM) Analyses have been recently pointed out as more efficient due to several practical advantages [3][4][5][6][7].

In this paper we propose to concentrate our efforts on a well-known Public-Key Algorithm [8], the RSA [9]. But the method could be easily transposed to any Public-Key Algorithm. These algorithms perform long and secret-exponent specific sequences of modular squaring and products. They are also characterized by different EM emission patterns. As a result, an adversary may typically try to disclose the secret exponent by identifying the repetitive patterns constituting a complete EM trace.

However, according to the hardware device on which is mapped the RSA algorithm, patterns associated to a squaring and a product might be really similar so that they cannot be distinguished. This is the case for the multiplexed architecture [10] considered in this paper (see Fig. 1). As a result, more powerful SCA must be applied to determine the secret exponent. For example, by applying differential or correlation analyses. However this requires spending a lot of time to acquire and process EM traces. One solution is to apply demodulation based techniques to increase the signal to noise ratio, as suggested, in [11][12] to disclose remaining leakages.

Within this context our first contribution is an experimental amplitude-demodulation based technique to enhance simple EM analyses. Unlike to [11][12], the proposed procedure is fully numeric and does not require additional equipments such as an expensive TEMPEST receiver [3] or a phase shifter [11]. Moreover, our solution does not involve any CPU-time consuming statistical tool such as the Mutual Information technique [12]. Our second contribution is a set of experimental procedures to analyze left-to-right square-and-multiply [14], square and multiply-always [15] and Montgomery powering ladder [16] implementations of the RSA. Note that [12] addresses only the case of the left-to-right square-and-multiply. Finally, our last contribution is an experimental evidence that the considered architecture [10] is robust to simple EM analyses but can still be attacked if memory accesses remain unprotected.

The paper is organized as follows. In section II, basics about amplitude modulation are recalled to point out where the remaining leaking information must be tracked in the frequency domain. Section III presents the proposed amplitude demodulation-based technique while the section IV presents the robustness evaluation results of three algorithms and the relevant target architecture details. Finally, a conclusion is drawn in section V.

II. Searching Information in the Frequency Domain

In this section, some basics about amplitude modulation (AM) are reminded. These basics provide guidelines to search the leakage information in the frequency domain.

For the sake of simplicity, we first consider that the leakage $L(t)$ is a sinusoidal signal of frequency f_L and amplitude A_L. Because, this leakage may appear at each clock cycle in the worst case scenario, the frequency f_L is necessarily lower than the clock frequency f_{clk}. As a result, the EM signal collected over several tenths or hundreds of clock cycles must have similar properties than the signal S_{am} resulting from the amplitude modulation of the $L(t)$ by the clock signal of frequency f_{clk} and amplitude A_{clk}:

$$S_{am}(t) = A_{clk}(t)A_L(t)\cos(2\pi f_L t)\cos(2\pi f_{clk} t) \qquad (1)$$

According to the Fourier Transform properties, the leakage appears at two different positions (both close to the clock

978-3-9810801-8-6/DATE12/©2012 EDAA

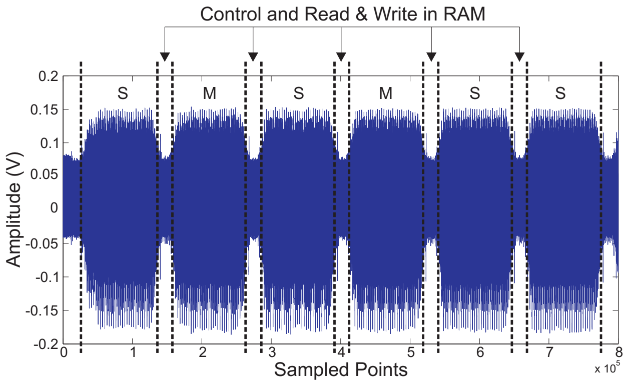


Fig. 1. A typical EM trace collected above the multiplexed architecture of [10]. S and M denotes squares and multiplies.

frequency) in the spectrum as shown by Equation 2 that gives the power spectral density of S_{am}.

$$\Phi(\omega) = \frac{1}{2}F(\omega + \omega_c) + \frac{1}{2}F(\omega - \omega_c) \tag{2}$$

If considering a sinusoidal signal to model the clock signal may be acceptable while working with circuits operating above the GHz, this is clearly not the case for smartcard products that typically operate at few tenths of MHz. The clock signal should therefore be modelled by a square signal. Similarly, the leakage resulting from the switching of CMOS gates or structures should be better modelled by a composite signal. According to these observations, the carrier involved in the amplitude modulation is given by:

$$p(t) = \frac{4}{\pi} \sum_{n=1,3,5\ldots}^{\infty} \frac{1}{n} \sin(2\pi n f_{clk} t) \tag{3}$$

In the case of a leakage signal $f(t)$ composed of three harmonics f_{L1}, f_{L2} and f_{L3}:

$$\begin{aligned} f(t) \quad = \quad & A_{L1}\cos(2\pi f_{L1} t) + A_{L2}\cos(2\pi f_{L2} t) \\ & + A_{L3}\cos(2\pi f_{L3} t) \end{aligned} \tag{4}$$

S_{am}, becomes:

$$\begin{aligned} S_{am}(t) \quad = \quad & f(t)\sin(2\pi f_{clk} t) + \frac{1}{3} f(t)\sin(2\pi 3 f_{clk} t) + \\ & \frac{1}{5} f(t)\sin(2\pi 5 f_{clk} t) \end{aligned} \tag{5}$$

As shown in Fig. 2, the leakage appears at different frequencies in the spectrum. The interesting point is that the low frequencies composing the leakage are necessarily closed to the carrier harmonics. Therefore, an adversary may search for leaking information in narrow bandwidths located close to all harmonics of the clock signal falling into the bandwidth of its SCA platform.

Once have been identified the frequency bandwidth(s) of interest, we first remove all uninteresting harmonics using bandpass filtering (with 5^{th} order Butterworth filters) and then we recover the leakage by demodulating the filtered traces.

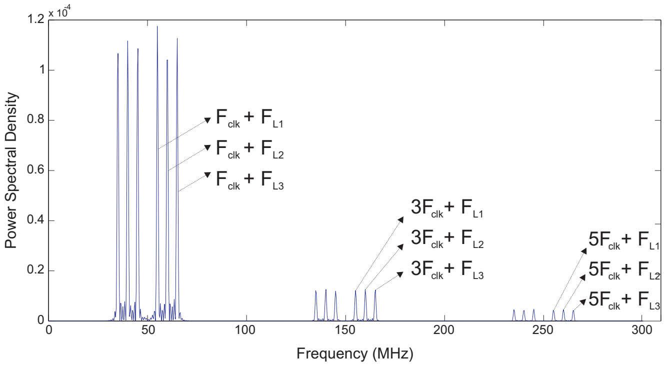


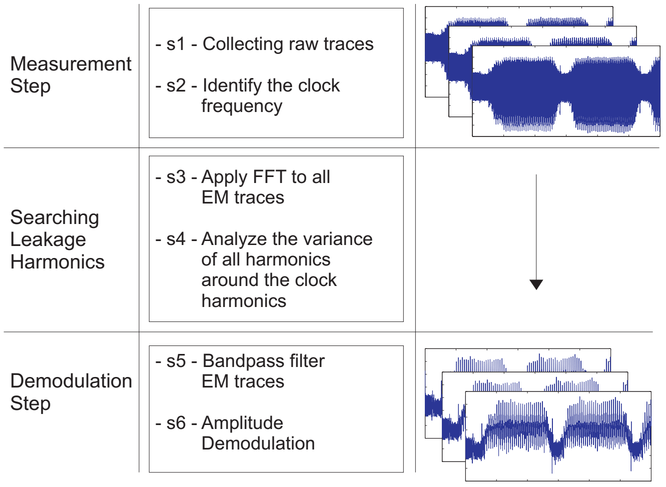
Fig. 2. Power spectral density of a signal modulated by a square wave carrier. The information signal has frequencies f_{L1} =5MHz, f_{L2} =10MHz and f_{L3} =15MHz.

This finally results in the experimental setup described in Fig. 3. It is worth observing that the only information required to apply it is the value of the clock frequency; information that can easily be obtained from basic EM analyses.

III. Application to Different RSA Implementation

From the side-channel point of view, finding the bandwidth(s) containing the leakage, or part of it, is a crucial step to obtain interesting results with the above procedure. This search could be in practice tedious or even infeasible depending on the considered cryptographic algorithms.

However, in the case of the RSA algorithm, finding the bandwidth(s) containing the leaking information is straightforward. It simply stands on computing the Power Spectral Densities (PSD) of modular squaring and products and then in identifying harmonics that significantly differs from one PSD to the other. Note, this is feasible in the case of RSA implementation because the most significant bit of a modular exponentiation is always *one* and, consequently, the three main computations steps are known (e.g., square/multiply/square,


Measurement Step

- s1 - Collecting raw traces
- s2 - Identify the clock frequency

Searching Leakage Harmonics

- s3 - Apply FFT to all EM traces
- s4 - Analyze the variance of all harmonics around the clock harmonics

Demodulation Step

- s5 - Bandpass filter EM traces
- s6 - Amplitude Demodulation

Fig. 3. Experimental procedure to disclose tiny leakage hidden in noise.

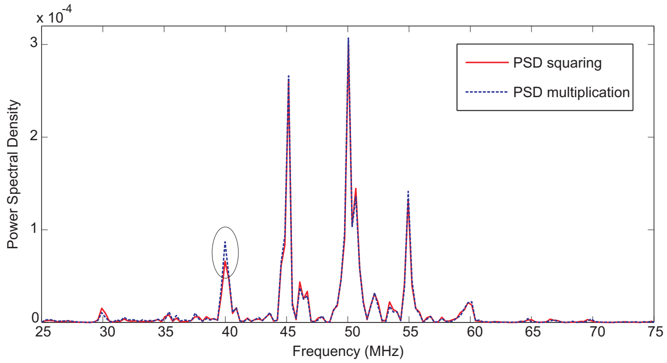


Fig. 4. Power spectral density of a modular squaring and a modular multiplication execution.

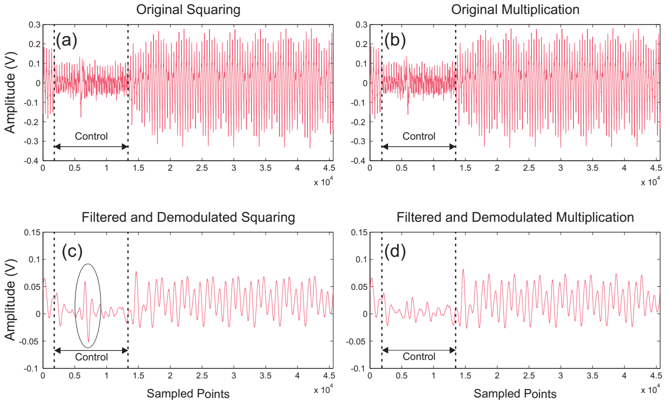


Fig. 5. Original (a) squaring and (b) multiplication traces and filtered and demodulated (c) squaring and (d) multiplication traces.

for the left-to-right square-and-multiply algorithm).

In the case of the architecture [10] considered herein, we have computed and characterized, around the clock frequency ($\approx$ 50 MHz), the power spectral densities of a modular squaring and a modular product. It is important to note that we consider during the computation the short time window related to the activity of the control logic and RAM accesses (see Fig. 1). Fig. 4 gives both averaged PSD obtained with 40 EM traces (20 modular squaring and 20 modular products). Since significant differences appeared around 40 MHz, we therefore retained two narrow bandwidths for bandpass filtering because the double-side band property of the AM. The first was centered at 40 MHz and the second at 60 MHz. Finally, we demodulated the filtered traces to obtain the results reported in Fig. 5.

Comparing Fig. 5a and 5b, which present a time frame of a modular squaring and a modular multiplication, respectively, one may conclude that no information leaks during the modular operations. Indeed, the right parts of Figures 5a and 5b are really similar. Note this was an expected advantage of using the multiplexed architecture of [10] in which the same Montgomery Multiplier [13] performs the modular squaring and products.

Comparing now the left part of Figures 5c and 5d that corresponds to the time window during which RAM accesses and control operations are performed, one may clearly identify a spike differentiating a square from a multiply. As a result, one may conclude that the proposed filtering and demodulation procedure is able to visually disclose a remaining and tiny leakage.

IV. Electromagnetic Analyses

EM traces were collected with a measurement platform composed of: an oscilloscope (bandwidth: 2.5 GHz; sampling rate: 40 GS/s), a low-noise amplifier (48 dB gain and 1 GHz bandwidth), a 500 μm probe, a motorized stage, a FPGA board and a PC to control the whole measurement setup. A cartography process, for acquiring EM averaged traces in 34 × 34 points (x,y) over the die area of the chip, was made to analyse the above multiplexed architecture and more precisely to determine the (x,y) coordinates at which the EM are strongly data dependent. This was done using the method proposed in [4].

For EM measurements, the multiplexed RSA architecture was set up with two different word sizes: 16 and 32 bits. The goal was to analyse the word size effect on the leaking syndrome. The arithmetic operations in the target RSA architecture are sequences of Montgomery modular multiplications and are performed in a multi-precision context. Therefore, bigger the word size, more gates are switched per clock cycle when performing such single-precision operations, and, consequently, more EM is emanated. Similarly, bigger the word size of the read and written words during the memory accesses, the more information is expected to leak during to RAM accesses.

Keeping this idea in mind, we applied our amplitude-demodulation based technique, designated by Simple and Demodulated Electromagnetic Analysis (SDEMA), on the three implemented modular exponentiation algorithms to evaluate if the exponent bits, manipulated by three different RSA implementations, may be recovered by processing a single EM trace as it can be done by applying a SPA or SEMA on an unprotected RSA.

Note that, in the left-to-right square-and-multiply case, our analyses aimed at identifying, as usual, square and multiply executions to directly disclose the exponent bits. In the case of the square-and-multiply always algorithm, our analysis aimed at finding the occurrence of *dummy* multiplications characterizing exponent bits with a zero value. Finally, to analyse the Montgomery powering ladder, we aimed at disclosing the squared operand of squares executions, which reveals the exponent bit value.

Our analyses adopt the sliding window method, which allows computing the differential trace between a reference frame (sampled points of a modular multiplication related to a known exponent bit) and frames related to an unknown exponent bit. The sliding window method was applied considering only one averaged EM trace (20 trials).

In the following sections, we provide the results and procedures for the SDEMA attack, applying our amplitude-demodulation based technique, on the three implemented modular exponentiation algorithms. The target architecture performs modular exponentiations through successive Montgomery multiplications which the input message is first converted into the Montgomery domain. The operation *Mont(X, Y, N)* always gives the result $XY2^{-n} mod\ N$, where n is the key (exponent) size and 2^{-n} is the Montgomery constant. This constant is removed from the final result by a last call to $Mont()$ having A and 1 as input parameters. The steps 1 and 2 in Alg. 1, 2 and 3 are precomputed and stored in memories.

To highlight the efficiency of our method, we also provide results for the simple EM analysis, without filtering and amplitude demodulation processes.

A. SDEMA on Left-to-Right Square-and-Multiply

In the left-to-right square-and-multiply (Algorithm 1), the first exponent bit is always *one* (MSB), because this is a downward method. The sliding window method employs the following steps:

1) apply the bandpass filtering and amplitude demodulation on the considered EM trace;
2) the three first computations are square-multiply-square;
3) select the sampled points related to one of these first computations to be a reference window frame;
4) compute the differential trace between the reference window frame and frames related to unknown exponent bits.

Fig. 6 and 7 show the results for the SDEMA and SEMA attacks on the left-to-right square-and-multiply implementation, respectively, considering the word sizes of 16 and 32 bits. We can observe that the sampled points related to the control activity (C) are on the left side of the dotted line.

Unlike the SEMA results for the two considered word sizes, the SDEMA attack discloses the exponent bits by displaying, in the left part of the differential EM traces, shapes that are specific to either square or multiply. These differences (spikes) appear because different operands are read for the square and multiply operations. Note, however, that the difference is quite reduced in case of 16 bits architecture contrarily to the 32 bits architecture.

Algorithm 1: Left-to-Right Square-and-Multiply

```
Input: $m$, $e$, $N$, $R = 2^n mod\ N$ (pre-computed).
Output: $c = m^e\ mod\ N$
1. $A = Mont(1, R^2, N)$;
2. $B = Mont(m, R^2, N)$;
3. for $i = n - 1$ to 0 do
4.   $A = Mont(A, A, N)$
5.   if $e_i = 1$ then
6.     $A = Mont(A, B, N)$
7.   end if
8. end for
9. $c = Mont(A, 1, N)$
```

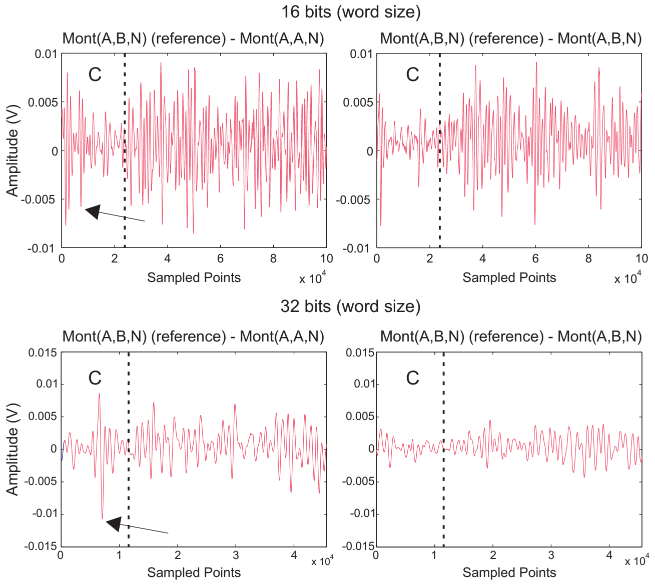


Fig. 6. SDEMA analysis on the left-to-right square-to-multiply implementation.

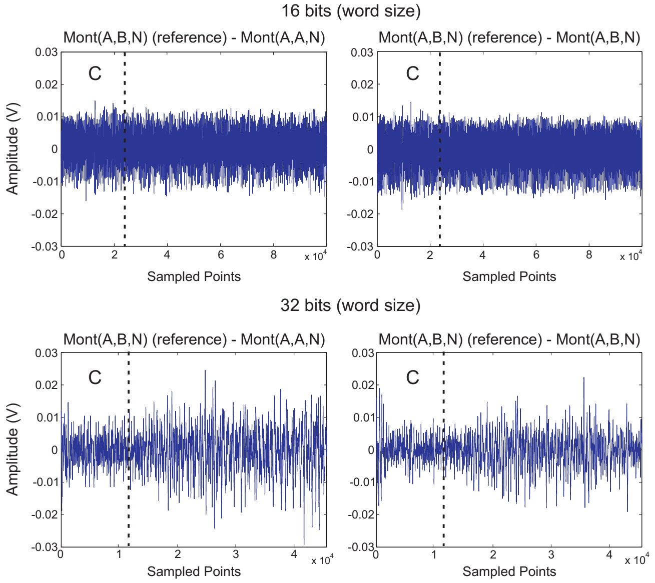


Fig. 7. SEMA analysis on the left-to-right square-to-multiply implementation.

B. SDEMA on Square-and-Multiply Always

The square-and-multiply always (Algorithm 2) is a downward method, therefore the first exponent bit is always *one* (MSB). For every exponent bit interpretation, one can find a square followed by a multiply. If the exponent bit is *zero*, the modular multiplication is a *dummy* operation. So:

1) apply the bandpass filtering and amplitude demodulation on the considered EM trace;

Algorithm 2: Square-and-Multiply Always

```
Input: $m$, $e$, $N$, $R = 2^n mod\ N$ (pre-computed).
Output: $c = m^e\ mod\ N$
$A = Mont(1, R^2, N)$;
$B = Mont(m, R^2, N)$;
for $i = n - 1$ to $0$ do
    $A = Mont(A, A, N)$
    if $e_i = 1$ then
        $A = Mont(A, B, N)$
    else
        $X = Mont(A, B, N)$ (dummy)
    end if
end for
$c = Mont(A, 1, N)$
```

2) the second computation is not a *dummy* modular multiplication;
3) select the sampled points related to this effective multiply computation to be a reference window frame;
4) compute the differential trace between the reference window frame and all frames of multiply computations, to identify the *dummy* multiplications.

Fig. 8 and 9 illustrate the results for the SDEMA and SEMA attacks, respectively. The results are presented for 16 and 32 word sizes.

The target multiplexed architecture does not store the result of *dummy* modular multiplications. So, if the differential trace presents highest amplitudes during the time window (C) related to the control activity, one may conclude that we are subtracting a effective multiply from a *dummy* operation. Otherwise, it means that the target modular multiplication frame is not a *dummy* operation.

C. SDEMA on Montgomery Powering Ladder

To apply the proposed SDEMA analysis on the Montgomery Powering Ladder implementation, we considered the algorithm according to the downward method (Algorithm 3). Thus, the first exponent bit is always *one* and, therefore, the two first executions are a multiply followed by a square. The remainder modular multiplications are always a multiply followed a square, however the squared operand of the modular squaring execution (A_0 or A_1) indicates the bit value of the exponent. So:

1) apply the bandpass filtering and amplitude demodulation on the considered EM trace;
2) the second computation is a modular squaring, having A_0 as squared operand;
3) select the sampled points related to this modular squaring computation to be a reference window frame;
4) compute the differential trace between the reference window frame and the frames related to modular squaring computations, to identify the modular squaring having A_1 as squared operand.

When applying SDEMA and a sliding window method it is possible to reveal the exponent by observing the different RAM accesses for the remainder modular squaring executions. The results are presented in Fig. 10. Fig. 11 shows the results for simple EM analysis.

Indeed, when a time frame of a modular squaring execution $A_1 = Mont(A_1, A_1, N)$ is subtracted from $A_0 = Mont(A_0, A_0, N)$, spikes appear at the beginning and at the end of the differential trace, indicating that different operands have been read and stored. On the other hand, if we are sub-

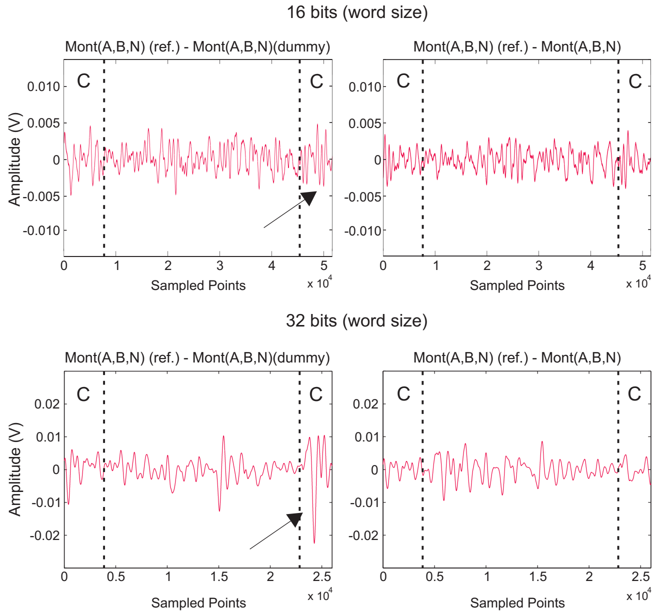


Fig. 8. SDEMA analysis on the square-to-multiply always implementation.

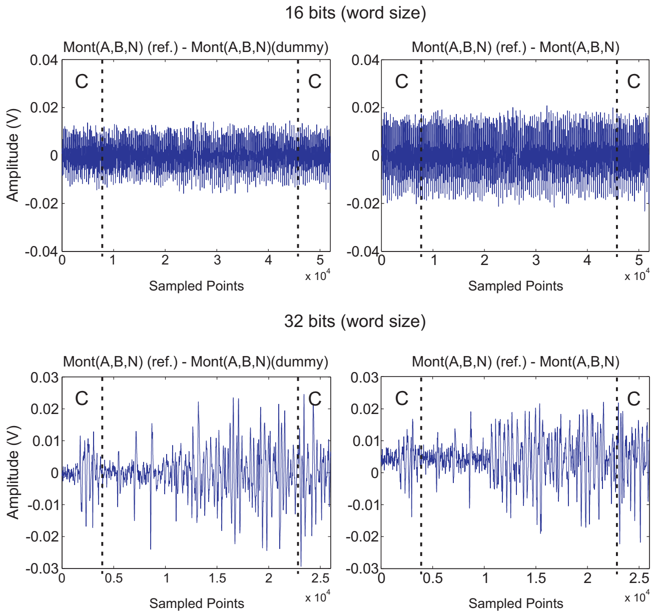


Fig. 9. SEMA analysis on the square-to-multiply always implementation.

Algorithm 3: Montgomery Powering Ladder

```
**Input:** $m$, $e$, $N$, $R = 2^n mod\ N$ (*pre-computed*).
**Output:** $c = m^e\ mod\ N$
$A_0 = Mont(1, R^2, N)$;
$A_1 = Mont(m, R^2, N)$;
**for** $i = n - 1$ **to** 0 **do**
    **if** $e_i = 1$ **then**
        $A_1 = Mont(A_0, A_1, N)$; $A_0 = Mont(A_0, A_0, N)$
    **else**
        $A_0 = Mont(A_0, A_1, N)$; $A_1 = Mont(A_1, A_1, N)$
    **end if**
**end for**
$A_1 = Mont(A_0, 1, N)$
```

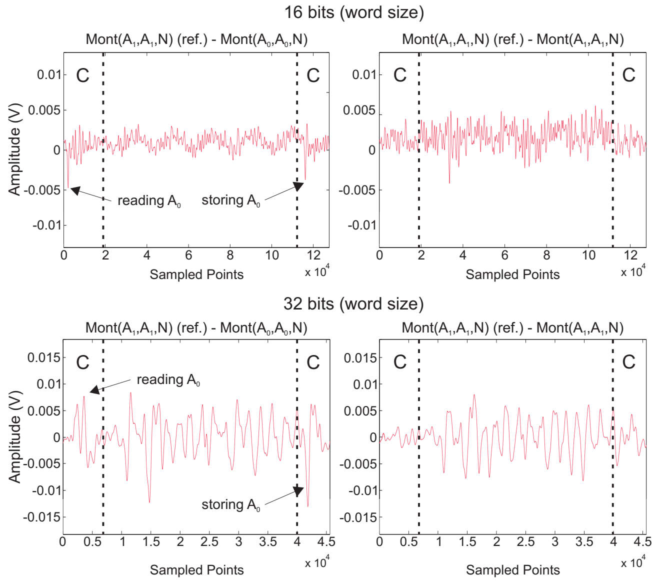


Fig. 10. SDEMA analysis on the Montgomery Powering Ladder implementation.

tracting modular squaring executions with the same squared operands, the differential trace is smoothed.

V. Conclusion

This paper presented an amplitude demodulation technique (SDEMA) to enhance simple electromagnetic analysis. The analysis based on amplitude demodulation and bandpass filtering processes was described. Its application to three different modular exponentiation methods configured with two different word sizes was presented. The results obtained with SDEMA demonstrate its efficiency on simple protected RSA methods like square-and-multiply always and Montgomery powering ladder over only one averaged trace (20 trials), because the filtered and amplitude demodulated traces reveal weakness of the control elements (RAM memories access, multiplexers addressing). The obtained results also highlight the need for specific RAM memories able to perform dummy write and read operations to enhance the robustness of the multiplexed architecture.

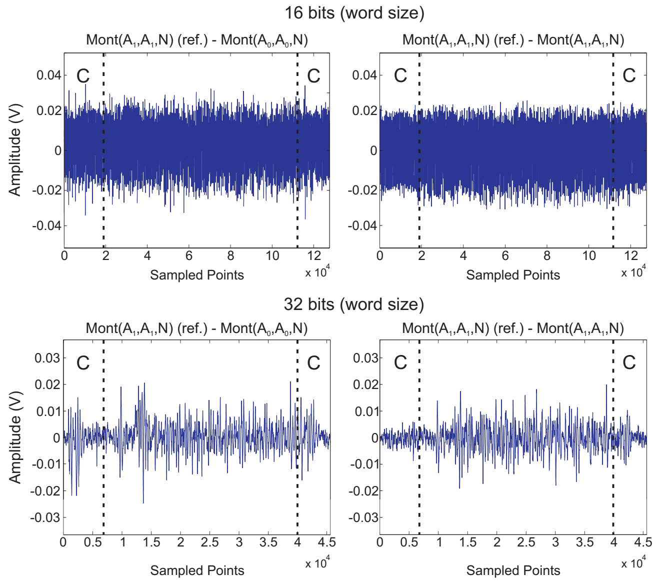


Fig. 11. SEMA analysis on the Montgomery Powering Ladder implementation.

References

[1] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *CRYPTO*, pages 388–397, 1999.

[2] E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In *CHES*, Springer-Verlag, pages 16–29, 2004.

[3] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The em side-channel(s). In *CHES*, Springer-Verlag, pages 29–45, 2002.

[4] A. Dehbaoui, V. Lomné, P. Maurine, L. Torres, and M. Robert. Enhancing electromagnetic attacks using spectral coherence based cartography. In *VLSI-SoC*, pages 135–155, 2009.

[5] K. Gandolfi, C. Mourtel, and F. Olivier. Electromagnetic analysis: Concrete results. In *CHES*, Springer-Verlag, pages 251–261, 2001.

[6] D. Réal, F. Valette, and M. Drissi. Enhancing correlation electromagnetic attack using planar near-field cartography. In *DATE*, pages 628–633, 2009.

[7] L. Sauvage, S. Guilley, and Y. Mathieu. Electromagnetic radiations of fpgas: High spatial resolution cartography and attack on a cryptographic module. *TRETS*, 2(1), 2009.

[8] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644 – 654, nov. 1976.

[9] L. Adleman. R.L Rivest, A. Shamir. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21:120–126, 1978.

[10] G. Perin, D. G. Mesquita, and J. B. Martins. Montgomery modular multiplication on reconfigurable hardware: Systolic versus multiplexed implementation. In *International Journal of Reconfigurable Computing*, volume 2011, 2011.

[11] T. Kasper, D. Oswald, and C. Paar. Em side-channel attacks on commercial contactless smartcards using low-cost equipment. In *WISA*, pages 79–93, 2009.

[12] O. Meynard, D. Réal, F. Flament, S. Guilley, N. Homma, and J.-L. Danger. Enhancement of simple electro-magnetic attacks by pre-characterization in frequency domain and demodulation techniques. In *DATE*, pages 1004–1009, 2011.

[13] P. L. Montgomery. Modular Multiplication Without Trial Division. *Mathematics of Computation*, 44:519–521, 1985.

[14] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone. The Handbook of Applied Cryptography. Boca Ranton, Fla.: CRC Press, 1997.

[15] J.-S. Coron. Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems. *CHES*, Springer-Verlag, pages 292–302, 1999.

[16] M. Joye, and S., M. Yen. The Montgomery Powering Ladder. *CHES*, Springer-Verlag, pages 291–302, 2002.