Extraction of the target signature for DPA/DEMA attacks using HHT and K-means clustering

Rodrigo Lellis, Adão A. Souza Júnior, Luciano Ludwig Loder
Instituto Federal Sul-Rio-Grandense - IFSul
Pelotas, Brazil
{nuevolellis, adaojr, lucianoloder}@gmail.com

Rafael Iankowski Soares
Universidade Federal de Pelotas - UFPel
Pelotas, Brazil
rafael.soares@inf.ufpel.edu.br

***Abstract*— Differential Power Analysis (DPA) is a very popular class of non-invasive side channel attacks. However, its effectiveness requires that the extracted power traces have a good temporal alignment. Several design countermeasures rely on the temporal misalignment of the stages in the cryptographic algorithm to protect against side channel attacks. For the attacker it is thus fundamental to identify the portion of the observed traces in which the processing occurs. This segment extraction stage of the attack is commonly based on a manual definition of a threshold. However, the use of threshold is problematic, since it depends on the observation of the traces besides being very sensitive to noise. This work proposes an attacks flow in which the extraction step is done automatically using the K-means clustering allied to the Hilbert-Huang transform. The flow was tested using traces with GALS architecture that provide both delay and frequency randomization. Results show that the process is effective and a presents a reduction of 56.51% in the average amount of traces required to carry out the attack in relation to the using only K-means clustering.**

Keywords*— DPA; DEMA; side channel attacks; cryptoanalysis; signal processing; power consumption; electromagnetic radiation.

I. INTRODUCTION

Side Channel Attacks (SCA) are a well-known class of attack proposed by Kocher [7]. These attacks correlate physical characteristics from cryptographic devices while they perform encryption or decryption with the input data. Among the physical characteristics exploited in the attacks are processing time, power consumption, electromagnetic radiation and even sound [2].

The most popular type of attack is Differential Power Analysis (DPA) [8] and Differential Electromagnetic Attacks (DEMA) [1]. The popularity of these attacks is due its non-invasive feature and its effectiveness. In this type of attack power consumption (or EM) signals from the encryption of distinct randomly generated plaintexts inputs are stored. After that, the processing segment of the traces is extracted and separated according to the type of switching caused by input data (low to high or high to low). Then, the average of each group of traces is calculated and its difference results in a differential trace. The correct key causes the highest peak of the differential trace [8]. This whole process requires that the power traces from each acquisition are as aligned as possible for the attack to succeed.

To prevent DPA and DEMA, several countermeasures are found in literature to reduce correlation among the acquired power (or EM) signal. One technique, called Random Delay Insertion (RDI) [12], introduces temporal misalignment between algorithm stages. Variation of the clock frequency also results in traces with a low correlation, according to [15]. Soares et al. introduces a Globally Asynchronous and Locally Synchronous (GALS) pipeline architecture that causes the simultaneous effects of random delay and clock frequency variation in the power consumption traces [14].

This kind of countermeasure has still some vulnerabilities. Loder at al. [11] propose an attack framework based on Phase Only Correlation (POC) to realign the traces. This approach has the disadvantage that the traces are divided by groups of clock frequencies, which makes many more traces to be necessary for the attack to succeed. Thanh-Ha Le et al. [9] performs the realignment of the traces by calculating the energy. To do this, it divides the trace into segments and then calculates the energy of each segment, condensing the information from all points of the segment into a single point. However, the authors do not discuss the size of the appropriate segment and how that is found for any given architecture. Tian [15] proposed a method to identify power consumption peaks and realign traces partially, but identifying the peak is very sensitive to noise. Lellis et al. [10] use an attack flow that consists of a step of extracting the target segment from the traces, a subsampling step and a trace energy calculation step. This method has the fragility of using a manual threshold set in its extraction stage, which as in the Tian method, is very sensitive to noise and requires the empirical observation of several traces.

The present work proposes to incorporate to an attack flow previously described [10], an automatic trace extraction step that identifies segments with processing and does not need manual definition of a threshold. This is accomplished through an unsupervised machine learning method, the K-means clustering [3], combined with the Hilbert Huang Transform [5] and to allow the calculation of instantaneous frequency.

This paper is organized as follow: Section II presents a theoretical overview and Section III describes the methodology of the present study. Experiments are explained in the Section IV. Results and conclusions are presented in Sections V and VI.

II. THEORETICAL OVERVIEW

A. K-means Clustering Method

K-means is an algorithm that seeks to divide a dataset into a given number of clusters that should be defined as an input parameter of the algorithm. Each cluster is represented by a centroid, i.e., by the center of all the data in the cluster. Each point is assigned to a cluster with the nearest centroid. After all points in the data set are distributed, the centroids are updated to include all points which are assigned to the cluster. This process occurs iteratively until a stopping criterion is reached. The algorithm is described in [3]:

i. Definition of the number of clusters, K;

ii. Random selection of K centroids;

iii. Calculation of the distance between each point and each of the centroids;

iv. Allocation of each point in the data set to the nearest centroid;

v. Update of the centroids according to the mean of the distances of the points in each cluster;

vi. Repeat items (iii) to (v) until the algorithm reaches the established stopping criterion.

It is worth mentioning that the distance in item (iii) can be calculated using different definitions. In this work, the Euclidean distance is used and calculated according to Equation (1):

$$d(p_i, c_i) = \sqrt{(p_{xi} - c_{xi})^2 + \left(p_{yi} - c_{yi}\right)^2} \qquad (1)$$

Where *p* represents the *i-th* point and *c* is the *i-th* centroid.

B. The Hilbert Huang Transform

The Hilbert-Huang transform is a method of time-frequency domain analysis that consists basically of two steps: the first results in empirically decomposing a given signal into a set of functions that best fits the signal in question you must first find a class of functions called Intrinsic Mode Functions (IMF) [5]. This step is called Empirical Mode Decomposition (EMD). In the second step, it is possible to obtain a representation in the time-frequency domain by calculating the Hilbert transform of each of the component of the first step. This stage is called Hilbert Spectral Analysis (HSA).

The IMFs must meet the following conditions:

i. The number of extrema and the number of zeros in the whole signal must be equal, or they may have a unit of difference;

ii. At any point, the average value of the envelope defined by local maxima and local minimums is zero.

IMFs are found through a process called sifting, which consists of the following steps:

i. Identify the extrema of the signal;

ii. Interpolate the extrema and get the upper and lower envelopes;

iii. Calculate the average of the envelopes;

iv. Subtract the mean of the signal;

v. Iterate in the residue;

Steps (i) to (iv) are repeated until the value found in (iv) has zero mean. After that, this value is considered IMF and step (v) applies.

The Hilbert transform applied to the IMFs found by EMD is calculated by the following convolution integral (2):

$$H(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \qquad (2)$$

Where *x(t)* is a given IMF and *P* is the principal value of Cauchy.

Therewith, we obtain the following analytical signal (3):

$$z(t) = x(t) + iH(t) \qquad (3)$$

that has as modulo the value given by the expression seen in (4):

$$a(t) = \sqrt{x(t)^2 + H(t)^2} \qquad (4)$$

and the phase given by (5):

$$\theta(t) = atan\left(\frac{H(t)}{x(t)}\right) \qquad (5)$$

From (5), the instantaneous frequency, i.e., the signal values in the time-frequency domain can be obtained by Equation (6):

$$\omega = \frac{d\theta(t)}{dt} \qquad (6)$$

In this work the Hilbert transform results are organized in descending order according to their power. The processing segment with the higher frequency will dominate the component with higher power giving an estimate to its instantaneous frequency.

III. METHODOLOGY

A. The Proposed Method

The present work proposes an automatic method to extract the relevant part of the power consumption traces for DPA/DEMA attacks, called target signature. Since in the attack flow of [10] the extraction step requires the definition of some parameters, such as a threshold. The use of a threshold is susceptible to noise, besides that the determination of such parameter occurs through the observation of the traces, which becomes infeasible for very different sets of traces.

To validate the extraction method proposed here an additional preprocessing step will be added a pre-existing the attack flow [10]. After this step, the extracted segments will be subsampled by a fraction that will result in vectors of the same length. Subsequently, the energy calculation will be performed to improve the alignment between the segments, and finally, the DPA/DEMA attack will be executed on the resulting traces.

Therefore, the complete attack flow performed in this work consists of the following parts: 1) Extraction, 2) Subsampling, 3) Energy Calculation and DPA/DEMA attack.

1) Extraction of the target segment in the traces of power consumption

The purpose of this step is go through a trace of power consumption and identify the signature corresponding to the processing of the device, discarding the remaining as noise. The detection defines the start and end points of the signature, thus allowing its extraction. This is illustrated in Figure 1.

The extraction process consists in: a) Perform the Hilbert-Huang Transform (HHT) on the traces*;* b) Perform the K-means in the transformed traces to find the starting point of the extraction and c) Run the K-means algorithm on the trace amplitude to find the end point of the extraction.

a) Perform the Hilbert-Huang Transform on the trace:

Before applying the transform, in order to increase the efficiency of the instant frequency detection and avoid the effects of higher frequency interferents found in the set, the original trace was filtered with a low-pass Butterworth filter.

Then the IMFs related to the traces through the EMD process were found. We can see the result of this process in Figure 2.

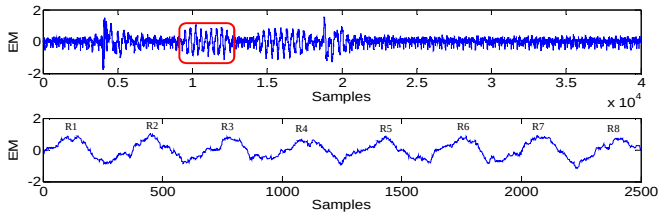

Figure. 1 Extracting the target segment from power consumption trace.

The next step is calculating the energy of each IMF to find the one with the highest power. HHT is performed in this component, resulting in the trace in the time-frequency domain, seen in Figure 3. One can see the execution areas of the cryptographic algorithm highlighted, in Figure 3.

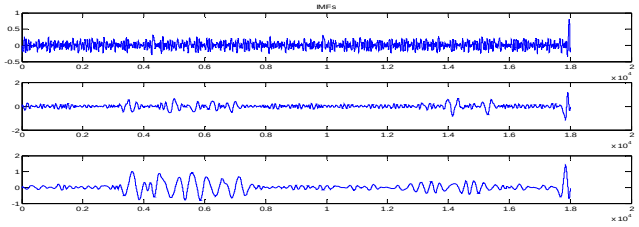

Figure. 2 IMFs 4, 5 and 6 from power consumption trace.

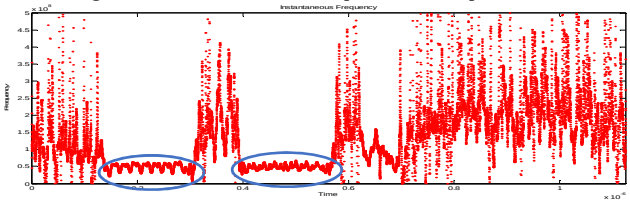

Figure. 3 Instantaneous Frequency from power consumption trace.

b) Perform the K-means in the transformed traces to find the starting point of the extraction

In this step, the number of clusters (value of K) was defined as two. One will be considered as containing the samples that correspond to the execution of the algorithm within the trace and the other will be discarded as periods were no relevant processing of the algorithm was performed (i.e. noise).

Clustering the instant frequency give good results to find the starting point of the segment. However, the end of the segment is not found with this method.

c) Run K-means on amplitude traces and find the end point of the extraction

At this point, again the K-means algorithm was executed, but this time on a function of the amplitude of the traces and not more on the frequency domain. To improve the efficiency of the K-means algorithm, a previous transformation was performed on the traces. This transformation consists of taking the third power of the segment values, thus increasing the difference between segments with and without processing and improving the detection ratio. *The end of the segment is taken as the largest value of the cluster which was defined to have valid processing*. The result of clustering can be seen in Figure 4.

2) Subsampling

The extracted segments have different sizes, even when no countermeasures are applied. This is due to small clock variations during encryption. Since traces were acquired with a rather high sampling rate and in order to the traces to have the same size, they are subjected to a subsampling step. This procedure also filters the segments eliminating aliasing [4].

3) Energy Calculation

According to the method proposed in [9] to align traces, the traces must be divided into segments and the energy of each segment is evaluated. In [9] is defined that the segment size should be large enough to cover the eventual temporal displacements, although it does not define a value for this size. However, in [10] a study evaluated the impact of segment size on DPA/DEMA attacks. Based on these previous results, the present work performs energy using segments with 200 samples, that correspond to half the cycle for the clock frequency of the set of traces used in the experiments [10]. It is noteworthy that the subsampling step decreases the number of samples also reducing the computational effort of the DPA/ DEMA attacks.

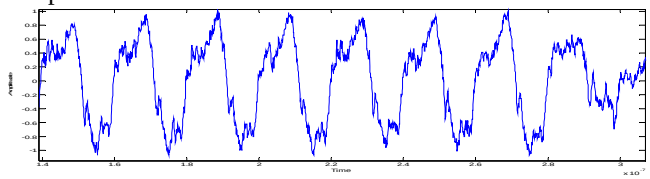

Figure. 4 An example of segment extracted from power consumption signals using the proposed automated method.

B. Experiments

A set of 100,000 EM traces was acquired from a GALS pipeline architecture with two stages implemented in hardware and prototyped on the Xilinx Spartan3 FPGA. In this architecture, each stage executes 8 rounds of the Data Encryption Standard (DES) algorithm [14]. For this work, the architecture operates at 50MHz frequency without countermeasures. However, as mentioned earlier, some variations around this operating frequency are present in these traces resulting from the execution, as well as certain time shifts known as jitter.

The traces of power consumption were acquired at a rate of 20G Samples/s. Since the clock frequency is 50MHz, one clock cycle corresponds to 400 samples. This justifies the calculation of energy with segments of size 200 of samples, which corresponds to half the clock cycle, as seen in [10].

DPA/DEMA defines the first round of DES as the main target of the attack meaning that during the extraction step, the first stage of execution of the algorithm is a crucial part of the signature and must be correctly extracted from the trace. Two different approaches for segment extraction are tested: Using only K-means on the amplitude signal to find both the beginning and the end of the traces; and with the initial point of the extraction using HHT and K-means. We also compared the results for the K-means only method using two different algorithm implementations: in R and in MATLAB. After extracting the different length segments, the final subsampling step is performed using the shorter trace as a reference for the final trace length. This step of the attack flow uses the function *resample()* which applies an anti-aliasing FIR (Finite Impulse Response) filter [13] before resampling.

The outcoming traces is then submitted to the energy calculation step. Finally, the DEMA attack is performed on the resulting traces. Evaluation metric for the vulnerability is defined as the minimal number of traces that is necessary to reach a successful attack.

IV. RESULTS AND DISCUSSIONS

Table I summarizes the experiments performed with the use of K-means implemented in R Language [6], K-means implemented in MATLAB and the method proposed in this work. Also, Table I shows the results obtained with the Threshold method [10] for comparison. Both K-means in R Language and MATLAB and the Threshold method have the steps of sub-sampling and calculating the energy in common in the attack flow. Column *Average* of Table I shows, on average, the quality of alignment obtained by the methods presented and represented by the minimal number of traces for a successful DEMA attack.

Observing the Table I, the number of traces required to recover the sub-keys corresponding to the SBOXs 5 and 8 stand out from the rest. This phenomenon is likely caused by problems during the acquisition of the EM traces. For this reason, the average number of traces to recover the sub-keys does not take into account the SBOXs 5 and 8.

Table I. Attack results from different experiments on the traces of power consumption with frequency of clock of 50MHz.

Method	SBOX1	SBOX2	SBOX3	SBOX4	SBOX5	SBOX6	SBOX7	SBOX8	Average
K-Means R Language	2266	10285	22816	3103	N/C	9528	3067	N/C	8510.83
K-Means MATLAB	1846	4246	7123	5716	N/C	7490	4824	N/C	5207.50
HHT and K-Means	966	984	6275	5365	38686	6313	2305	N/C	3701.33
Threshold	167	1016	1611	1443	5386	2359	1299	N/C	1315.83

According to Table I, we can see that clustering through the K-means algorithm had different results for the two implementations, and the implementation in MATLAB presented a reduction of 38.81% in the average amount of traces in relation to its implementation in the R Language [6]. This result motivates us to use scripts written in MATLAB for the experiments.

We also note that the use of the HHT allied to clustering to find the starting point of the segment presented a 28.92%. reduction in the average number of traces needed to a successful attack in relation to the K-means only extraction method. Moreover, the proposed method presents a reduction of 56.51% when compared to the K-means extraction implemented in R.

It may be noted that although the use of Threshold proposed by [10] presents a greater reduction in the amount of traces in relation to the method proposed in this work, the use of a threshold is problematic because it depends on observation of the traces and is very sensitive to noise.

V. CONCLUSIONS

For DPA/DEMA attacks to be successful, power consumption traces must be well aligned. This allowed countermeasures based on misalignment of traces to be proposed. In contrast, vulnerabilities in these countermeasures were found through pre-processing steps in the traces. Among the countermeasures found in the literature, we can cite [10] that is based on a threshold to perform the extraction of the target signature of the traces. This implies a fragility of the method, since the use of threshold beside needing the observation of the traces, what can be infeasible is very sensitive to noises. Thus, the present work proposes an automatic solution for this flow stage through clustering with the K-means algorithm. An improvement is carried out in this process by using the Hilbert-Huang transform, obtaining the instantaneous frequency of the traces before the clustering to find the starting point of the extraction. The remainder of the attack flow from [10] is used after the extraction step, thus passing the segments by a step of subsampling and then calculating the energy. These latter steps reduce the number of tracer points, which reduces computational effort in DPA/DEMA attacks.

We can see from the obtained results that the method proposed here for the target segments extraction step presents a reduction of up to 56.51% in relation to the simple application of clustering by the amplitude of the traces using K-means implemented in the R Language, which confirms the potential of the method to incorporate the DPA/DEMA attack flow. Future work could apply the same attack flow to implementations of the cryptographic algorithm in devices equipped with countermeasures, such as RDI and random clock frequency variation.

REFERENCES

[1] AGRAWAL, D.; ARCHAMBEAULT, B.; RAO, J. R. and ROHATGI, P. "The EM Side-channel(s): Attacks and Assessment Methodologies," in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, Aug, 2002, pp. 29-45.

[2] BACKES, M., DURMUTH, M., GERLING, S., PINKAL, M., and SPORLEDER, C. “Acoustic Side-Channel Attacks on Printers,” in *USENIX Security Symposium*, Aug, 2010, pp. 1-16.

[3] HARRINGTON, P. “Machine Learning in Action,”. in *Greenwich, CT, USA: Manning Publications Co.*, 2012.

[4] HAYKIN, S. and VAN VEEN, B. *Sinais e Sistemas*. Bookman, 2001.

[5] HUANG N, SHEN Z, LONG S, WU M, SHIH H, ZHENG Q, YEN N, TUNG C, LIU H; “The Empirical mode decomposition and the Hilbert sepctrum for nonlinear and non-sationary time series analysis,” 1998, Proc R. Soc. Lond. A, 454, 903-995.

[6] JUNIOR, P. F. “Um estudo sobre a aplicação de técnicas de clusterização para extração da assinatura alvo do traço em ataques DPA” *Undergraduate Thesis UFPel*, 2018, 49p.

[7] KOCHER, P.; "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," in *Annual International Cryptology Conference on Advances in Cryptology (CRYPTO)*, Aug, 1996, pp. 104–113.

[8] KOCHER, P. C.; JAFFE, J.; JUN, B. “Differential Power Analysis,” in *19th International Cryptology Conference on Advances in Cryptology*. Santa Barbara, USA: (CRYPTO'99) Springer-Verlag. 1999. pp. 388-397.

[9] LE, T. H. et al. "Efficient Solution for Misalignment of Signal in Side Channel Analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr, 2007, pp. 257-260.

[10] LELLIS, R.; SOUZA, A.; SOARES, R. “An energy-based attack flow for temporal misalignment coutermeasures on cryptosystems,” in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, May, 2017, pp. 1-4.

[11] LODER, L.; SOUZA, A.; FAY, M. and SOARES, R. "Towards a Framework to Perform DPA Attacks on GALS Pipeline Architectures," in *Symposium on Integrated Circuits and Systems Design (SBCCI)*, Aug - Sep, 2014, pp. 1-7.

[12] LU, Y.; O’NEILL, M. and MCCANNY, J. "FPGA Implementation and Analysis of Random Delay Insertion Countermeasure against DPA," in *International Conference on Field-Programmable Technology (FPT)*, Dec, 2008, pp. 201-208.

[13] MATLAB, "MATLAB Support Documentation," Available in: https://www.mathworks.com/help/signal/ug/resampling.html. [Accessed: 31 - May - 2018].

[14] SOARES, R.; CALAZANS, N.; MORAES, F.; MAURINE, P. and TORRES, L. "A Robust Architectural Approach for Cryptographic Algorithms Using GALS Pipelines," *IEEE Design & Test of Computers*, vol. 28, no. 5, pp. 62 –71, Sep-Oct, 2011.

[15] TIAN, Q. and HUSS, S. A. "On Clock Frequency Effects in Side Channel Attacks of Symmetric Block Ciphers," in *International Conference on New Technologies, Mobility and Security (NTMS)*, May, 2012, pp. 1–5.