Bulletin of the *Transilvania* University of Braşov
Series I: Engineering Sciences • Vol. 6 (55) No. 2 - 2013

A PRACTICAL APPROACH TO POWER TRACE MEASUREMENT FOR DIFFERENTIAL POWER ANALYSIS BASED ATTACKS

C.L. PIȚU[1,2] R. CÂMPEANU[1]

Abstract: *This paper addresses the issue of power supply noise in differential power analysis based attacks on microcontrollers. Using a mixture of analogue filters with a special powering technique, the noise effect of the power supply, on the acquired data, is minimized. More, the data acquisition and therefore the DPA are simplified by synchronizing the sampled data with the system clock the microcontroller runs at. This approach provides a simple method of recording power traces for DPA attacks. In the second part of the paper a comparison of classic measurement versus the one suggested here, is presented. This paper presents practical tips for doing exploratory DPA in order to detect security vulnerabilities.*

Key words: *AES, DPA, Arduino, AVR.*

1. Introduction

Differential Power Analysis (DPA) is a form of side channel attack in which the attacker studies the power consumption of a digital device in order to gain knowledge about its functioning mode and the data computed at a certain moment in time. This technique was first presented more than 10 years ago [2] as an academic research method and has evolved ever since into a powerful tool for engineers as well as for attackers. Originally DPA attacks were directed against smartcards [3] with more recent attacks on microcontroller devices and FPGAs.

Differential power analysis basically allows an attacker to compute the intermediate values of a cryptographic computation by statistically analyzing data collected from multiple cryptographic operations [1], [3], [5]. This data consists of the power consumption of the analyzed device (power trace) and the plain text data over which the cryptographic operations are performed.

A DPA attack uses just a digital oscilloscope and a post-processing algorithm, implemented in software. Depending on how many power traces have been recorded and how complex the processing algorithm is, the attack can last from a few hours to days.

As power traces are usually noise tainted, noise filtering methods have been developed. These methods use signal processing based algorithms. Another option is to filter the power supply induced

[1] Control Systems and Information Technology Dept., *Transilvania* University of Braşov.
[2] Electronics Design Dept., Siemens Corporate Technology Romania, Braşov, România.

noise in order to reduce the post-processing time and speed up the attack.

Today there are many DPA based techniques presented in technical literature and research papers. Most of these are difficult to be applied in practice and require a great effort to set up and execute. Current problems of DPA attacks are:

- power traces are tainted with noise;
- acquired power traces require large storage space;
- attack times are long.

All these issues influence the performance of a DPA attack in terms of needed resources and processing time.

A basic DPA attack set-up consists of a measuring setup consisting of a low noise power supply, a digital oscilloscope and the attacked device (or IC). Through a standard communication interface data is fed into the device. A GPIO pin is used as a trigger for the power trace recordings.

Today, more than ever, electronics are interconnected, therefore fulfilling the IoT dream. One of the most prominent IoT platforms is Arduino. Arduino is both hardware and software and is built around the well known AVR chip family from Atmel.

As communication is usually encrypted, a plethora of encryption algorithms have been ported to the Arduino platform. One of the best crypto-libraries is provided directly by Atmel and focuses on the well known AES (Advanced Encryption Standard). AES is characteristic for its 10 computing rounds which usually can be easily seen on an oscilloscope.

DPA attacks on AES implementations are nothing new but the effort needed to execute them is still considerable.

The presented acquisition method has been developed to decrease to effort on the post-processing algorithms in order to speed up the attack and reduce the needed computing resources. Using a dedicated software framework, called Power Analysis Toolkit, and a combination of analogue noise reduction techniques and digital pre-filtering mechanisms, the current paper presents a practical approach to doing DPA attacks.

The paper is organized as follows. Section II presents the filtering methods used and the powering scheme proposed. Data acquisition approaches and pre-filtering methods are presented in section III. A practical example is presented in section IV while section V concludes the paper.

2. Noise Reduction Techniques

Digital electronics are usually designed to be immune to a certain level of electric noise. As depicted in Figure 1 the noise can have a significant effect on the power trace.

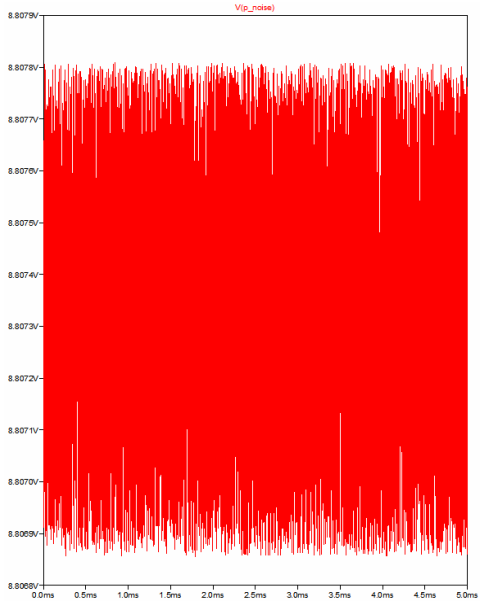

Fig. 1. *Power line noise*

As microcontrollers draw a small current, they are mostly powered over USB - interface used also for communication. Over a USB port a maximum current of 500 mA at 5V can be drawn. Although this approach seems practical it also raises the noise level

on the power lines as can be seen Figure 2. The noise on the USB power line is due to the switching power. Another powering technique is to use an external power supply. These can be filtered or unfiltered and are usually standard transformer-based.

For performing a DPA attack there is a special need for noiseless power lines. Therefore, in order to minimize the noise level two approaches are used.

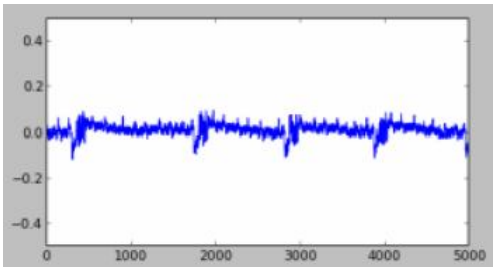

Fig. 2. *USB power line noise*

2.1. RC filter

The first and easiest filter to be applied is the classic RC low-pass filter. This filter has great results if the noise frequency is known. Using the already known equation:

$$f_{cutoff} = \frac{1}{2\pi RC}, \tag{1}$$

a simple filter can be calculated. In case the circuit is powered over USB, the noise level is above 10 kHz thus the RC filter can be successfully applied.

The only inconvenience about the RC filter is the voltage drop over the series resistor. This can cause stability problems of the digital circuit (brown out detection, execution instability etc.); therefore special care has to be taken to minimize this effect.

A typical RC filter contains no expensive components - just a resistor and a capacitor. The advantage of this filter is also its immunity to other components, being few inductive.

A practical solution to this problem will be presented in the following.

2.2. LC filter

The LC filter is much better suited for filtering a wider noise spectrum while causing no voltage drop over its components, like in the case before. The LC filter can be calculated, again, using the standard, already known formula:

$$f_{cutoff} = \frac{1}{2\pi\sqrt{LC}}. \tag{2}$$

This approach works best in case of an external power supply, e.g. a transformer-rectifier-stabilizer type of supply.

LC filters are most used on AC noise sources and have high efficiency. The disadvantage of this filter is its sensitivity to other components due to its high inductivity.

This two filtering methods were selected due to their simplicity and ease of use. Both method are very effective and can be easily applied in DPA measurement set-up's. The RC filter is most recommended do to the high availability of resistors and capacitors in a wider variety than inductors.

In order to perform a DPA attack, one need to record the power traces, thus the power consumption of the digital circuit has to be measurable and noise filtered. As depicted in Figure 3 a noise tainted power supply can be filtered up to a level where the noise has no effect on the measurement. As can be seen, the best results are obtained using a RC low-pass filter.

As previously stated the IoT platform of choice is Arduino. The actual circuit used in this platform is ATmega328, an 8 bit microcontroller which is manufactured by Atmel Inc. In order to perform a DPA attack on this circuit, some guidelines are:

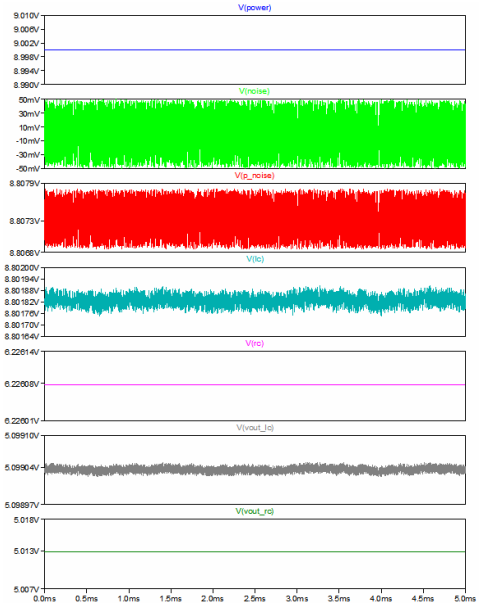

Fig. 3. *LC and RC noise filters*

- turn off the Brownout detection - this causes a power drain similar in size to the CPU;
- turn off all unneeded modules - this reduces the consumption to almost zero if the CPU is not running;
- do not use power management functions - these might cause false readings and thus wrong power traces;
- do not use timers running in parallel to the main computing thread - instead use sequential logic;
- do not use interrupts - these cause high power drain which makes it difficult to perform a DPA attack.

Another aspect to consider is the trigger pin which signals the execution of the operation of interest to the outside world. This pin is usually used to trigger the digital oscilloscope to recording the power traces. Theoretically any pin (GPIO) can be used but practical measurements show that toggling a pins state causes a big spike visible in the power trace. This spike is difficult to filter thus a better approach is needed.

The ATmega328 has two power supply pins - one for the CPU and communication modules and one dedicated to the ADC. As only the power consumption caused by the CPU and the memory accesses are of interest, the ADC module can be powered separately, on the second power supply pin (AVcc). This allows switching the state of any pin in PORT C without having an effect on the measured power consumption.

The circuit, still Arduino compatible, is powered from an external (non switching mode) power supply. The board is equipped with a standard RC low-pass filter and a LC filter which are used to remove the noise from the power line - the filters are selectable by jumpers. Both filters are tuned for a non-switching power supply with an output voltage of 9V.

Using a TL431 as a regulator, the circuit is powered at 5V/100 mA. The powering scheme inserts a shunt resistor both in the Vcc as well as the GND lines. Depending on the measuring apparatus, the desired configuration can be selected via jumpers. The shunt resistor inserted in the GND line is the easiest approach if a standard oscilloscope and probes are used. For measurements using the shunt resistor in the Vcc line differential probes (or a non-floating oscilloscope) are needed.

Validation of the powering scheme is done both in simulations, using LTspice [8], as depicted in Figure 3, as well as in practice, using the developed board presented above and a digital oscilloscope (LeCroy WaveSurfer 424).

The circuit is developed on a test board (Figure 4) and is programmed using the standard Arduino IDE [7] and the AES library [6] provided by Atmel. No special considerations are taken into developing the software application running inside the Arduino platform. On power up, the ATmega328 first executes a boot loader. This enables the chip to be programmed over serial interface instead of using a standard ISP programmer.

On the computer side a Python script is used to trigger the cryptographic operations. A basic communication protocol is used in order to load the keys and plain text data into the microcontroller and trigger the cryptographic operation.

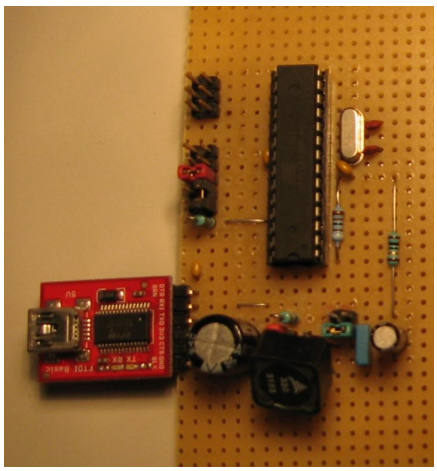

Fig. 4. *DPA board*

3. Data Pre-Filtering and Acquisition

Data acquisition is performed using a digital oscilloscope. Each power trace holds one cryptographic operation, from start till finish, as depicted in Figure 5. The memory accesses for the S-Box look-up table as well as the 10 computing rounds are clearly visible. The obtained power traces can be quite big in size (~3 MB) thus a few thousand traces easily sum up to a few GB of data. Storage is probably not an issue today but having such large amounts of data need large processing times. There are few solutions which can be applied in order to reduce the amount of needed storage on the one hand side and to speed up the processing on the other hand side.

Digital oscilloscopes have a buffer-like memory which stores "an acquisition". This usually means more than the needed time window during which the operation of interest is performed. Also, the power trace is tainted with the static power drained by the digital circuit - this can be considered as noise and has only a negative impact on the needed power trace. Therefore two solutions are presented which, when used together, offer good results with minimum effort.

3.1. Pre-Filtering Technique

One of the main problems in recording power traces for DPA based attacks is the static power consumption of the circuit under analysis. This consumption is caused by the circuit's internal structure and can be considered noise from the point of view of the acquisitioned data. Due to the design style of the circuit (e.g. clock tree) there is no direct method to reduce or eliminate the static power drain. The most reliable option to reduce the static power noise is to put the device into sleep but this would prevent it from executing the desired operation.

From the point of view of the power trace post-processing there are a number of techniques available but most of them require high computing power and time. A cheap solution, given the existence of a high performance digital oscilloscope, is a two step approach to "reduce" the static power of the analyzed circuit. This solution consists of:

- recording the static power consumption of the circuit and storing it in the RAM (due to speed considerations) of the oscilloscope;
- performing an average of the recorded static power consumptions (a rule of thumbs states that about 100 static power traces are enough to obtain a usable average);
- then, for each recorded power trace of the circuit:

- record the power trace and store it in RAM;
- subtract the averaged static power consumption from the power trace;
- store the result on a non volatile

memory and displaying the result of the oscilloscopes display.

Using this approach, the "average static noise" of the digital circuit is removed from each power trace, leaving only the information which holds the data of interest - the power consumption which is directly proportional to the processed data.

3.2. Post-Filtering Technique

The stored power traces now hold the information of interest. Still, each power trace contains much more information than needed. The time window during which the cryptographic operation takes place is much shorter (about ½) of the entire recorded power trace. Due to its construction, the oscilloscope save as a power trace the entire content of its internal buffer, from the trigger event until the buffer is full. This unnecessary data does not hinder the DPA attack, but causes it to take longer time.

There are two possible solutions in this case. One solution would be to use the traces directly in a differential power analysis. This would for work but the needed processing time would be longer and probably at some point in time the results would yield a false positive. A second approach is to crop the traces to the needed length. In order to cut the unneeded information, the trigger signal is used as a reference. As depicted in Figure 5 the trigger signal goes high when the cryptographic operations are performed. Using a python script the power traces can be parsed and, using the trigger signal as a reference, cut down to the useful length. This still means that each power trace is parsed and processed offline, before the actual DPA attack is performed.

In order to benchmark the two techniques, 1000 power traces have been recorded. Each trace is one cryptographic operation, more specifically one encryption.

Fig. 5. *AES power trace*

Using the initial approach, meaning without any optimizations at all, the 1000 traces needed 3.3 GB of storage space (roughly 3.4 MB/trace). Using the optimization mechanisms presented, the 1000 traces needed only 0.97 GB of storage space.

The difference might not be very large but in case of 10000 traces or more the difference becomes visible.

4. A Practical Example

Using the above mentioned inputs and optimizations, a basic DPA attack was performed. The test bench consisted of a modified Arduino Uno board equipped with an ATmega328 microcontroller running at 16 MHz. The microcontroller was preprogrammed with a boot loader which allows the chip to be programmed over the serial interface instead of using an ISP programmer. The application attacked is based on the AES library from Atmel and a communication protocol implemented over the serial interface. This allows the external loading of keys and plain text data as well as reading the operation results and comparing them to the expected one.

On the computer side, a Python script was used to load the keys and plain text data and to trigger the cryptographic operation.

From the functional point of view, after the key and data was loaded, the microcontroller was idle for a few clock cycles in order to make the difference in power consumption more visible.

The digital oscilloscope was set to trigger mode which allowed it to store one trace at a time in RAM, perform the pre-filtering operations and then store it to a non-volatile memory. Due to the processing time needed by the oscilloscope, a delay of ~40 ms was set between the cryptographic operations. The power consumption was measured using a differential probe, on the Vcc line shunt (10 Ω) without any analog amplification or filtering.

Using these setup 10000 power traces were recorded and prepared for the DPA attack.

The actual DPA attack consisted of a modified correlation scenario as presented in [4] and was also implemented in Python. The results of the attacks, in terms of performance, are presented in Table I: All the tables have to be created using the utility Table. The measuring units will be put down within square brackets, in the head of the table.

Duration of Various Operations Table 1

Operation	**Duration**	
	Initial	**With improvement techniques**
Acquisition	5,5 hours	7 hours
Recovering first key byte	5 hours	1 hour

This results in a difference of more than 4 hours in processing time. Both attacks were performed using the same computer as before, running on one core in a single thread.

5. Conclusions and Further Developments

This papers presents simple techniques which combined can lead to a more powerful power trace acquisition and thus DPA attack. Simple, standard filtering techniques combined have been applied in order to reduce the power supply noise and to allow the recording of a clean power trace. A modified powering scheme allows the microcontroller to signal its operation to the outside world by switching a pin's state, without affecting the actual measured power consumption.

Furthermore, a software filtering mechanisms allows for a cleaner power trace by reducing the static noise of the analyzed digital circuit and by cropping the unneeded part of the power trace which otherwise would prolong the DPA attack.

The presented technique does not require dedicated boards or special components - it has been designed to use available parts at a minimum price.

All presented software mechanisms were implemented, on the computer side, in Python, an easy to use but medium performance programming language. Another advantage of using Python is its portability thus allowing the applications to be run on Windows and Linux without any modifications at all. All used tools are either open source or freeware.

From the analysis of the power traces a second learning has been drawn - there is no visible difference in the power trace if the software running inside the microcontroller is optimized for speed or storage space or not optimized at all.

Further developments will be geared towards:

- optimizations of the acquisition algorithm;
- optimizations of the attack scenario;
- optimizations of the pre- and post-processing algorithms;
- more advanced DPA attack techniques
- protection techniques for software applications running in embedded systems.

The presented techniques can be easily ported to more platforms allowing for a better understanding of existing vulnerabilities and protection of the devices.

Acknowledgment

This paper is supported by the Sectoral Operation Programme Human Resources Development (SOP HRD), ID76945 financed from the European Social Fund and by the Romanian Government.

References

1. Boneh, D., Demillo, R.A., Lipton, R.J.: *On the Importance of Checking Cryptographic Protocols for Faults*. In: Proceedings of Advances in Cryptology - Eurocrypt '97, 1997, p. 37-51.
2. Kocher, P., Jaffe, J., Jun, B.: *Introduction to Differential Power Analysis and Related Attacks*. Available at: http://www.cryptography.com/dpa/technical. Accessed: 01-09-2013.
3. Mangard, S., Oswald, E., Popp T.: *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Graz. Springer, 2007.
4. Mangard, S., Oswald, E.: *Attack Scenarios*. Available at: http://www.dpabook.org/onlinematerial/matlabscripts/index.htm. Accessed: 01-09-2013.
5. Popp, T., Oswald, E., Mangard, S.: *Power Analysis Attacks and Countermeasures*. In: IEEE Design & Test of Computers **24** (2007) No. 6, p. 535-543.
6. *** *AES library*. Available at: http://www.das-labor.org/wiki/AVR-Crypto-Lib. Accessed: 01-09-2013.
7. *** *Arduino*. Available at: http://www.arduino.cc. Accessed: 01-09-2013.
8. *** *LTspice, Design Simulation and Device Models*. Linear Technology. Available at: http://www.linear.com/designtools/software. Accessed: 01-09-2013.