Bit-level Differential Power Analysis Attack on implementations of Advanced Encryption Standard software running inside a PIC18F2420 Microcontroller

Kealeboga Mpalane*
Department of Computer Science
North West University
Private Bag X2046, Mafikeng
2745, South Africa
Email: kmpalane@csir.co.za

H.D Tsague
Council For Scientific and Industrial Research
PO Box 395, Pretoria
0001, South Africa
Email: HDjononTsague@csir.co.za

Naison Gasela and E.M Esiefa
Department of Computer Science
North West University
Private Bag X2046, Mafikeng
2745, South Africa
Email: naison.gasela@nwu.ac.za and
esiefabukohwo@gmail.com

Abstract—Small embedded devices such as microcontrollers have been widely used for identification, authentication, securing and storing sensitive information. In all these applications, the security and privacy of the microcontrollers are of crucial importance. To provide strong security to protect data, these devices depend on cryptographic algorithms to ensure confidentiality and integrity of data. Moreover, many algorithms have been proposed, with each one having its strengths and weaknesses. This paper presents a Differential Power Analysis(DPA) attack on hardware implementations of Advanced Encryption Standard(AES) running inside a PIC18F2420 microcontroller.

I. KEYWORDS
Differential power analysis, Power attacks, AES, Microcontroller, Side channel attacks.

II. INTRODUCTION
Because cryptographic devices are widely in different applications that require strong security protection, Security of these devices are of crucial importance. For this reason, cryptographic device developers rely on cryptography to secure their data [1]. Consequently, cryptographic devices depend on cipher algorithms to ensure confidentiality and integrity of data. The goal of cryptography is to use cryptographic algorithms to protect data from unintended individuals by converting it into a non-recognizable form which cannot be read by anyone except the intended recipient. [2].

Although cryptographic devices have been known for protecting secret information, cryptanalysts are still able to break the security of most cryptosystems by studying and analyzing the information system in an attempt to recover its hidden characteristics [3], [4]. This happened after the introduction of Side Channel Attacks(SCA) in 1998 by Paul Kocher [5]. Since then, microprocessors have been targets of security attacks. [5] showed that Power Analysis Attacks(PAA) can reveal secret information of a device by using the leaked information. Microcontrollers are implemented using Complementary metal-Oxide Semiconductor (CMOS) technology. CMOS circuits consumes electric power by charging load capacitances every time they are switched [6]. This consumed power can be used as leaked information that attackers can use to break the system in the sense that the consumed power depends on the operations performed by the computing device [7], [8]. Examples of leaked information are electromagnetic radiations, power consumption, and timing information measurements leaks [9]. This information is used to obtain the secret key or other information stored on the device. There are two kinds of PAA, Simple Power Analysis(SPA) and Differential Power Analysis(DPA). They are both based on statistical methods established by Kocher et.al [10], [11]. DPA is much more powerful than SPA and more difficult to prevent [12], [13]. Most cryptographic systems use the implementations of AES algorithm because AES is believed to be mathematically strong [5]. However, they can be broken by using PAA. [8], [14] showed that power consumption measurements of a device measured while performing multiple algorithmic operations can be used to extract the secret key of AES implementation. DPA is the most used technique against cryptographic algorithms implemented in cryptographic devices [14].

This paper presents DPA attack in a PIC18F2024 running an implementation of AES algorithm which is used as a target for the attack. DPA will be used because it is most practical and economical physical attack [14]. It uses statistical methods to extract private information from the power consumption of the cryptographic device under attack [8] [5]. The remaining sections of the paper are organized as follows. Section 2 presents DPA techniques. Section 3 describes the method applied to acquire data and addresses the description
III. DPA EXECUTION

DPA attacks are the most popular type of power analysis attacks which is widely used to reveal the secret keys of cryptographic devices. DPA require a large number of power traces which are measured while the device encrypt or decrypt different data blocks. However, the secret key of the device can be revealed even if the measured traces are extremely noisy. The main advantage of DPA is that it doesn't require detailed knowledge about the device but adequate knowledge of the cryptographic algorithm executed by the cryptographic device is needed.

Figure 1 above illustrates the principle that DPA follows. Firstly, the attacker input known data (inputs) to the device under attack and measure its power consumption while it is performing encryption or decryption operations. Secondly, they calculate the hypothetical power consumption for all possible values of the secret key. Thereafter, they compared the hypothetical power consumption and the measured power consumption values. Lastly, the correct key byte is revealed by correlating the hypothetical power consumptions with the measured power consumption. In a successful attack, the correct key byte hypothesis will show a significantly high peak relative to other key bytes hypotheses.
Algorithm 1: Algorithm for performing DPA attack using Hamming-weight as a power model and DoM as a correlation method.

Input: Plaintexts, traces
Result: Results matrix of size 256*100002

1. for key byte position $b=0:16$ do
   2. get key byte position $b$ for each plaintext;
   3. Predict the intermediate values;
   4. for key guess $k=0$ to 256 do
      5. for plaintext $p=1$ to 1000 do
         6. Predict the power consumption;
         7. PowerConsumption = bitget(AfterSbox,bit);
      8. end
   9. for trace no. $i=1$ to 1000 do
      10. Correlate the predicted power consumption with the traces;
      11. Generate difference traces;
   12. end
13. end
14. return Results
15. end

It is believed that the power consumption of the target device depends on all the bits of the output byte at some moment in time. Hamming weight power model was used to generate the hypothetical power consumption. Finally, Difference of Means (DoM) method was used to measure the statistical dependency between the measured power consumption and the hypothetical power consumption. The results based on 1000 traces generated a Results matrix which generated graphs for every key byte hypotheses. The graph with the highest peak was taken as the correct key byte. As illustrated in Algorithm 1, line 7 was used to guess the instantaneous power consumption values for all the encryption runs for all key hypotheses, $bit$ is the target bit number, and bitget returns the bit value at position bit in the integer array AfterSbox. To correlate the hypothetical power consumption with the measured power consumption (Algorithm 1, line11), Difference of Means (DoM) statistical method was used and can be calculated as shown in (1.1 through 1.5).

$$\text{mean}_{1i,j} = \frac{1}{n_{1i}} \times \sum_{l=1}^{n} HT_{l,i} \times MT_{l,j}$$  \hspace{1cm} (1)

$$\text{mean}_{0i,j} = \frac{1}{n_{0i}} \times \sum_{l=1}^{n} (1 - HT_{l,i}) \times MT_{l,j}$$  \hspace{1cm} (2)

$$n_{1i} = \sum_{l=1}^{n} HT_{l,i}$$  \hspace{1cm} (3)

$$n_{0i} = \sum_{l=1}^{n} (1 - HT_{l,i})$$  \hspace{1cm} (4)

Results = $\text{MEAN}_1 - \text{MEAN}_2$  \hspace{1cm} (5)

HT denotes hypothetical power consumption, MT denotes measured power traces and $n$ denotes the number of power traces used for the attack. The inputs used for the algorithm are:
Plaintexts: AES inputs bytes of size (1000x16)
Traces: power traces of an AES microcontroller implementation of size (1000x10002) corresponding to plaintexts.

VI. Attack Results

The experimental results of Figure 3 shows key byte plots for bit 1, 2, 3 and 4 of byte 1 and as depicted by the figure, all the four bits revealed the key byte with bit1 revealing most of the information about the key. Although bit 2 didn’t reveal much, the information was enough to reveal the key byte. From the figure, it can be concluded that different bits in the same micro-controller register leaks different amount of information.

Fig. 3. Plots for first four bits of Byte1.

Figure 4 shows all the correct bits that revealed the key byte for byte1. All the bits except bit 7 revealed the key byte. The plots are overlapping, hence other plots are invisible.
Figure 5 shows a plot for the incorrect key byte for byte 1. The target bit for this byte was bit 7. The bit didn’t reveal any information and this might be because leakage of each bit of the target intermediate value is different and each bit leaks independently.

VII. CONCLUSION

In this paper we have applied a DPA attack on AES software implementations running on PIC18f2420 micro-controller. Main contribution of this work is highlighting that different bits of the target intermediate values can be used to reveal information about the secret key. Experimental results showed that using different target bits leads to indecisive results. For future work, it is necessary to define countermeasures suitable for DPA attacks to protect the secret key.

VIII. ACKNOWLEDGEMENT

The authors would like to thank: Council for Scientific and Industrial Research (CSIR), Department of Science and Technology (DST), and North West University (NWU) for their financial support.

REFERENCES


