I. J. Communications, Network and System Sciences. 2008; 1: 1-103
Published Online February 2008 in SciRes (http://www.SRPublishing.org/journal/ijcns/).

Efficient DPA Attacks on AES Hardware Implementations

Yu HAN[1], Xuecheng ZOU, Zhenglin LIU, Yicheng CHEN
Department of Electronic Science & Technology, Huazhong University of Science & Technology, Wuhan, P.R.China
E-mail: [1]husthyt@gmail.com

Abstract

This paper presents an effective way to enhance power analysis attacks on AES hardware implementations. The proposed attack adopts hamming difference of intermediate results as power mode. It arranges plaintext inputs to differentiate power traces to the maximal probability. A simulation-based AES ASIC implementation and experimental platform are built. Various power attacks are conducted on our AES hardware implementation. Unlike on software implementations, conventional power attacks on hardware implementations may not succeed or require more computations. However, the method we proposed effectively improves the success rate using acceptable number of power traces and fewer computations. Furthermore from experimental data, the correlation factor between the hamming distance of key guesses and the difference of DPA traces has the value 0.9233 to validate power model and attack results.

Keywords: Security, AES, Differential Power Analysis (DPA), Power Model, Correlation Factor

1. Introduction

The security in mobile applications [1] is of crucial importance because a large number of nodes may be exposed in a hostile environment. And if only one node is captured by attackers, the impact to the whole network can be devastated. Therefore, various cryptographic services required for these applications involve not only solutions for data protection but also self-implementation concerns. Mobile nodes are usually equipped with hardware coprocessors which are used to perform security protocol. If they are captured by attackers, side-channel information leakages, such as timing, power consumption and electromagnetic radiation, may be monitored for cryptanalysis. Among them, differential power analysis (DPA) [2] poses a serious threat to the security of different cryptographic implementations because it is practical, non-invasive, and easy to repeat.

Power analysis attacks exploit the correlation [3] between the data and the instantaneous power consumption of cryptographic devices. As this correlation is usually very small, statistical methods should be used to exploit it efficiently. In a power analysis attack, an attacker first creates a hypothetical power model of the cryptographic device at a very abstract level. In practice, each cryptographic algorithm designed operates only small parts of the secret key, called subkey, at certain period. Thus, the attacker can write a simple computer program that executes the algorithm at that period. The program calculates the intermediate result of this part for all possible subkey guesses. These values allow for predicting the power consumption, which is related to the inputs of cryptographic algorithms and subkey guesses. Next, the attacker feeds the same inputs to the real cryptographic device and measures its power consumption. Then the attacker correlates the predictions of the power model with real power consumption. For all wrong key guesses, the predictions will not correlate with the real measurements, but for the correct key guess, there will be a visible peak for the power analysis traces. In order to set up the correlation, predictions from different power models and statistical methods must be tested.

AES [4] is a new symmetric block cipher standard, which was issued by the National Institute of Standards and Technology (NIST) on November 26, 2001. AES has special particularities suitable for area- and power-constrained applications. Hence, the secure AES implementation can greatly affect the nodes in severely resource-constrained networks. AES is a round-based symmetric block cipher and can be implemented efficiently on all kinds of platforms. The standard key size is 128 bits. But for some applications, 192 and 256-bit keys can be supported as well. The round consists of four different operations, namely, *SubBytes*, *ShiftRows*, *MixColumn*, and *AddRoundKey*. Each operation maps a 128-bit input state into a 128-bit output state. The state is represented as a 4×4 matrix of bytes. The number of rounds depends on the key size. For a 128-bit key (AES-

128 is our concern), the round starts with a single *AddRoundKey* operation followed by 9 identical computation rounds. And, a slight difference is that the final round has no *MixColumns* operation. Figure 1 shows an AES-128 encryption diagram, and more details can be found in [4].

In this paper, we conduct a successful DPA attack on an AES hardware implementation to examine its security. The remainder of the article is organized as follows. We review related work in section 2. Section 3 addresses the principle of our presented attack. A simulation-based experimental environment and power acquisition are presented in section 4. Experimental results and discussions are provided in section 5. Finally, we conclude in section 6.

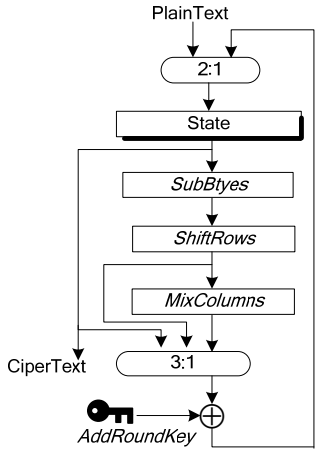

Figure 1. AES-128 encryption flow

2. Related Work

One of the most typical targets of power analysis attacks is the smart card, which is capable of performing secure computations. It consists of a (typically, 8-bit) processor, together with ROM, EEPROM, and a small amount of RAM. The cryptographic software basically operates on 8-bit data blocks because of the 8-bit architecture. In the original paper [2], P. Kocher et al. announced a DPA attack against the DES implementation in smart card microprocessors. In [5], T. S. Messerges et al. extended the research and presented experimental data and attack details. E. Brier [6] enhanced power analysis attacks using correlation factor between power samples and hamming weight of the handled data. All of these attacks have been extensively proved to be effective on symmetric and public-key encryption schemes in smart cards.

Contrasted to software implementations in smart cards, hardware implementations in FPGAs and ASICs are usually required for their ability to deal with high throughput. They allow parallel computing and have a more flexible architecture (as they are under the control of designers). Due to different processing behaviors and physical characteristics, a simple hamming power model can not be used to predict the power consumption of hardware implementations. Hardware implementations may leak less data-dependent power information to resist DPA attacks than software implementations. Recent publications [7, 8] show that DPA attacks has effectively defeated hardware implementations on cryptographic circuits as well. However, it requires a number of power measurements and a high computational complexity. To retrieve a secret key, an attacker must know more implementation details to deduce more precise power models. And he has to perform more complex statistical analyses to pre-process power data.

An important improvement has come with the appearing of high-order DPA [9]. This type of attacks generalizes DPA attacks by simultaneously considering multiple samples that correspond to several intermediate values within the same power trace. Template attack is presented as a new variant of power analysis attack. According to [10], this is the strongest form of side channel attack possible in an information theoretic sense. However, these new attacks require a deeper knowledge of the experimental device and more time consuming to mount. In this sense, it is therefore much less general. We have no comparison with attack results of these new methods in this paper.

According to above discussions, our work aims to improve first-order DPA attacks by developing a simple attack strategy against AES hardware implementations with fewer computations and more generality.

3. Principle of Improved DPA Attack

Current power analysis attacks [2][5][6][7][8] exploit the fact that the power consumption of a device executing a algorithm depends on the intermediate results handled. In these DPA attacks, random plaintext inputs and hamming model are used. Further, we assume that the differences between two different power measurements at the same sampling time are also related to the differences of the intermediate results at least to a certain degree. Thus, we deduce an improved power model with hamming difference of intermediate results, not hamming weight or hamming distance power model. In addition, partial plaintext inputs are fixed to set up DPA traces.

Power attacks can be divided into single-bit DPA and multi-bit DPA. In a single-bit DPA attack, a certain bit of intermediate result is predicted. It is used to split the power measurements into two sets, of which the means are computed and subtracted. For multi-bit DPA, multiple bits of intermediate results are predicted. In the two contexts, we have to verify the peak of bias signal by observing DPA traces. It is often subjective. The CPA is presented when a correlation factor between the outputs of power model and real power traces is shown. Correlation factors can be directly compared in different

CPA traces at cost of computational complexity. Our improved DPA approach overcomes these drawbacks in the following analyses.

3.1. Improved Power Model

We suppose that at time t an intermediate result, $I(x, t, k)$, which is an n-bit word, only depends on plaintext x and key k. Therefore, the general hamming weight model [6] based on a zero reference state for power consumption can be defined as

$$P(t) = aH[I(x,t,k)] + b. \quad (1)$$

Here a denotes a scalar gain between the hamming weight H and the instantaneous power consumption $P(t)$, and b is a hardware-dependent constant. Considering a plaintext x_1 with corresponding intermediate result I_1, we can obtain $P_1(t) = aH[I_1(x_1,t,k)] + b$. Similarly, when another plaintext x_2 results in the opposite intermediate result I_2, the corresponding instantaneous power consumption is $P_2(t) = aH[I_2(x_2,t,k)] + b$. Then, we can get the maximal difference of the power model in an absolute value:

$$\begin{aligned} &|\Delta P(t)| = |P_1(t) - P_2(t)| \\ &= \alpha |H[I_1(x_1,t,k)] - H[I_2(x_2,t,k)]| = \alpha \times n. \end{aligned} \quad (2)$$

where n is 128 for AES. This means that the power consumption of the whole circuit tends to be maximal. There is still the possibility even though an 8-bit subkey is concerned.

SubBytes is the sole nonlinear operation of AES, which consumes a majority of the area and power of AES. In addition, any intermediate result that occurs after *MixColumns* depends on 32-bit of the round key. This lead to a large number of subkey guesses needed to be tested, which is impractical. Furthermore, the subkey used for guessing should be the original key without key expansion. Accordingly, the intermediate results to predict power consumption target the output byte of the initial *AddRoundKey* or the output byte of following *SubBytes*. Each of them is a function of the plaintext byte and corresponding subkey guess. If the first byte subkey (denoted as *Ks*) of the first round key is targeted, and x_1, x_2 are the two corresponding plaintext bytes in two encryptions. We use the hamming difference of intermediate results under two different plaintext inputs as our power model, which shown as Figure 2. The predictions of the power model are given as follows:

$$P = aH(I_1) - aH(I_2). \quad (3)$$

For the target after *AddRoundKey*, I_1=*Ks* $\oplus$ x_1 and I_2=*Ks* $\oplus$ x_2. Whereas for the target after *SubBytes*, I_1=*SubBytes*(*Ks* $\oplus$ x_1) and I_2=*SubBytes*(*Ks* $\oplus$ x_2). We consider two cases: I_1=*0x00* and I_2=*0xff* regardless of which target. Here plaintext byte x_1 is for the former and x_2 for the latter. Thus, we can derive the corresponding plaintext bytes for each subkey guess *Ks* from the maximal prediction of the improved power model.

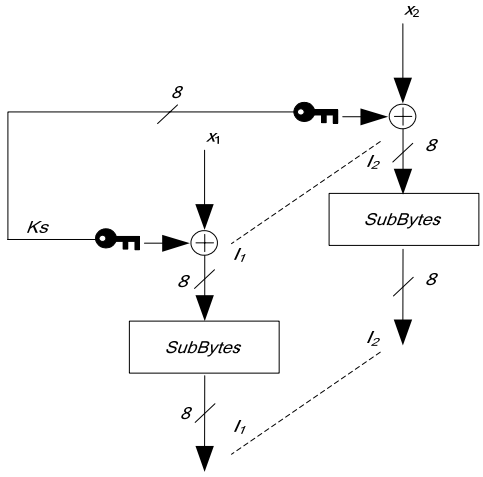

Figure 2. The improved hamming difference power

3.2. Improved DPA Traces

The generic power analysis attacks pay little attention to the choices of plaintexts during power sampling. The plaintexts are usually assumed random. But, for our improved attack, we first set plaintext bytes as x_1, x_2 obtained from the hypothetical power model, separately. Then, keeping other plaintext bytes random can result in a uniform distribution of remained partial bits of the intermediate results. And their influence on real power measurements can be eliminated if the number of random plaintext inputs is enough. Finally, for each subkey guess, we can prepare two plaintext sets as follows, each of which contains m plaintexts.

$$\begin{aligned} S_1(Ks) &= \{S_1[Ks,i] : (x_1, PTi[119:0]) \mid 1 \le i \le m\} \\ S_2(Ks) &= \{S_2[Ks,i] : (x_2, PTi[119:0]) \mid 1 \le i \le m\} \end{aligned} \quad (4)$$

where *PTi*[119:0] denotes 120 random plaintext bits, and they are the same in S_1 and S_2. Therefore, we can derive the plaintext inputs from the hypothetical power model for every subkey guess. Since there are 256 AES subkey guesses, the total plaintext number of two sets is 512×m.

For each subkey guess *Ks*, we perform AES encryptions using the above two plaintext sets in real power acquisition stage. And two power trace sets can be obtained, denoting $E(S_1(Ks), t)$ and $E(S_2(Ks), t)$. DPA trace between the two cases is presented as follows for the subkey guess *Ks*.

$$\Delta E(Ks,t) = \sum_{i=1}^{m} E(S_1[Ks,i],t) - \sum_{i=1}^{m} E(S_2[Ks,i],t) \quad (5)$$

when the correct subkey is assumed, a peak can be identified since it is obviously higher than other subkey guesses. In addition, we do not need to use the averaging statistics since the numbers of power traces in two sets are equal.

4. Power Acquisition in Simulation-Based Environment

There is now a demand to evaluate the DPA resistance of AES circuits. Thus, a typical hardware implementation of AES has been developed. The experimental conditions are shown in TABLE 1.

Table 1. Experimental Conditions

Description- languages	Verilog-HDL
Design technology	UMC CMOS 0.25μm 1.8v
Logic synthesizer	Synopsys DesignCompiler v200509
Power simulator	Synopsys PrimePower v200406sp1
PC spec.	CPU: Ultra SPARC Ⅱ 450MHz, Memory: 4GB, OS: Solaris9

Our AES implementation [11] is unfolded, including 16-byte registers for storing the intermediate results for each round of operation. All registers are set zeros when starting the encryption. The remainder combinatorial circuits perform *SubBytes*, *ShiftRows*, *MixColumns* and *AddRoundKey* operations. Each round operates during one clock cycle. And, we implemented the *SubBytes* unit with a classic GF architecture [12], which is especially suitable for resource-constrained applications.

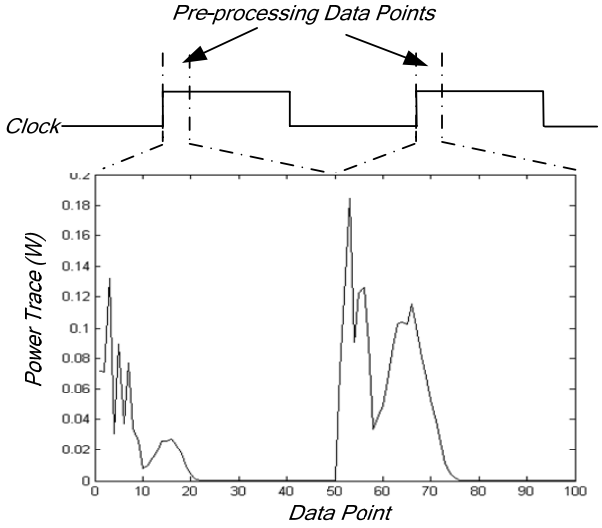

Figure 3. The power trace of the first two clock cycles measured for one AES encryption

During the power acquisition stage, the clock frequency applied to the system was 2.5 MHz and the sampling frequency was 1 GHz. The initial key addition operation occurred during the first clock cycle. And the result of this operation was written into register at rising edge of the second clock cycle. Hence, in the simulation-based power acquisition environment, we only measured the power consumption of the target period (the first two clock cycles of every encryption operation) during an AES encryption. 800 data points for two clock cycles during each encryption were acquired. One power trace is shown in Figure 3. The main component of power consumption is dynamic power consumption caused by data switching. The power trace in Figure 3 approaches the minimum from the 30th data point after the rising edge of the clock cycle, which is due to the critical path delay. To reduce computational complexity, a pre-processing technique was necessitated to eliminate the last 350 data points during every clock cycle. We considered those remaining 100 data points for two clock cycles as the instantaneous power consumption related to the target intermediate results.

5. Experimental results and discussions

5.1. Experimental results

We first conducted original power attacks on our AES implementation in the experimental environment, which involved singe-bit DPA, multi-bit DPA and CPA. We could not retrieve the right subkey from single-bit DPA and multi-DPA using 6000 power measurements. A CPA attack on the intermediate results of *AddRoundKey* revealed the correct subkey based on 4000 power measurements. As to the intermediate results of *SubBytes*, none of these attacks was successful. Successful CPA results are shown in Figure 4. The black plot denotes the

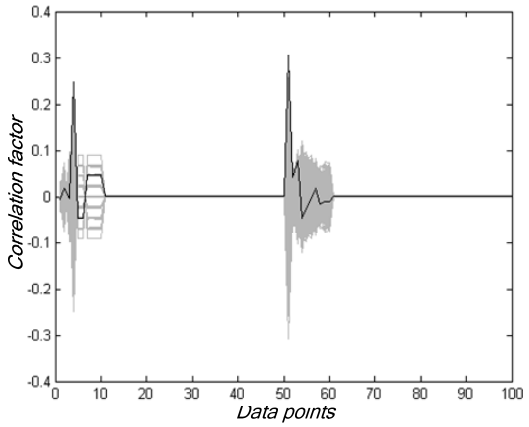

Figure 4. Correlation coefficients for 256 subkey guesses during two clock cycles

correlation coefficient traces for the correct subkey guess 0x74, whose peak is a little higher than the second highest point for incorrect subkey guess 0x16. In addition, an attacker can learn the moment of time when the instantaneous power consumption has a maximal correlation with the intermediate results of *AddRoundKey* by observing Figure 4. It also means that the AES hardware implementation has a maximal probability to leak data-dependent power at 5ns and 454ns during its encryptions. These two moments are closely related to the first *AddRoundKey* operation and the affine transformation of *SubBytes* operation, respectively. Thus, we conclude that these linear operations in the AES implementation result in more data-dependent power

leakages than other round operations.

We performed the improved power attack on the intermediate results of *AddRoundKey* and *SubBytes*, respectively. Like CPA, the correct subkey was extracted only for the intermediate results of *AddRoundKey*, and we took m = 10, denoting 5120 power measurements. Figure 5 shows the results of our improved attack. The peak for our improved DPA traces also occurs at about 4ns after starting AES encryption. That means the chosen plaintext inputs according to the improved power model generate

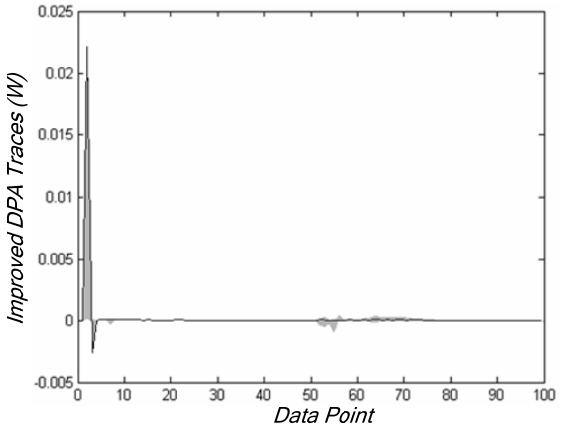

Figure 5. Improved DPA traces for 256

the maximal DPA signal when attacking the first *AddRoundKey*. In addition, the bias signal of the improved DPA trace between the peak and the second highest point is about 1.1mW, which provides a more effective comparison to the above results of CPA.

5.2. Discussions

From the results of our improved DPA, we compute the hamming distance *HD* between the subkey guess *Ks* and the extracted subkey *Kr* as following.

$$HD(Ks) = H(Ks \oplus Kr) \tag{6}$$

Since each power trace for attacking *AddRoundKey* contains 50 data points, we assume the sum of these samples as interesting DPA traces, computing as follows.

$$\Delta E(Ks) = \sum_{t=1}^{50} \Delta E(Ks, t) \tag{7}$$

Then, we also compute the difference $|\Delta P|$ between the two interesting DPA traces corresponding to subkey guesses as follows.

$$|\Delta P(Ks)| = \big||\Delta E(Ks)| - |\Delta E(Kr)|\big| \tag{8}$$

The correlation factor between *HD* and $|\Delta P|$ is 0.9233, which shows that $|\Delta P|$ has a perfect linear relation with *HD*. Therefore, we confirm that hamming power model is valid for our proposed method, and the correct subkey has been retrieved.

Single-bit DPA attacks have been successfully conducted on cryptographic software implementations in smart cards, but it is often not the fact for hardware implementations. It is due to their differences in processing behaviors and physical characteristics. Multi-bit DPA can upgrade the peak level by encrypting a large amount of random plaintexts. But sometimes it is impractical. CPA exploits the data-dependent power leakage in a statistical way, which has been proved to be effective on both software and hardware implementations. However, in comparison with our improved DPA, CPA uses the correlation coefficient which needs to compute a mass of expectations, variances and square roots. Only summing and subtracting are required in our improved DPA. Further, the proposed DPA peak can be verified more objectively than original DPA attack.

6. Conclusion

In this paper, an effective DPA method to retrieve the secret key from an AES hardware implementation is presented. Based on the improved power model, we can prepare the corresponding input plaintext bytes for every subkey guess in advance. In addition, our DPA traces can be built through simple summing and subtracting operations instead of complex statistical techniques. As the partitioning criterions of single- and multi-bit DPA are usually abstract and simple, these two DPA methods can not retrieve any useful information even with 6000 power measurements. Although the CPA attack can extract the right subkey based on 4000 power measurements, its computational complexity sometimes exhibits a bottle-neck. Compared with the methods mentioned above， our proposed DPA excels them in both effectiveness and computation requirements. Furthermore, the perfect linear relation validates the improved DPA attack and the power model by analyzing experimental data.

7. Acknowledgement

The research described in this paper has been supported by High technology Research and Development Program of China under grant 2006AA01Z226 and by Scientific Research Foundation of Huazhong University of Science and Technology under grant 2006Z001B.

8. References

[1] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, Spins: Security protocols for sensor networks, Wireless Networks, Vol. 8, pp. 521-534, 2002.

[2] P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, in Advances in Cryptology—CRYPTO 99. Heidelberg, Germany: Springer-Verlag, 1999, vol.

1666, Lecture Notes in Computer Science, pp. 398–412.

[3] J.M.Rabaey, A.Chandrakasan, and B.Nikolic, Digital Integrated Circuits, A Design Perspective, Second Edition, Prentice-Hall, Upper Saddle River, NJ, 2003.

[4] J. Daemen, V. Rijmen: AES Proposal: Rijndael, Document Version 2, 1999.

[5] T.S. Messerges, E.A. Dabbish, and R.H. Sloan. Examining Smart-Card Security under the Threat of Power Analysis Attacks. IEEE Transactions on Computers, 51(5), 2002.

[6] E. Brier, C.Clavier, F.Oliver: Correlation Power Analysis with a Leakage Model, In proceedings of CHES 2004, LNCS 3156, pp. 16-29.

[7] F.X. Standaert, S. B. Ors, J.J. Quisquater and B. Preneel Power analysis attacks against FPGA implementations of the DES. In Field Programmable Logic and Application. Heidelberg, Germany: Springer-Verlag, 2004, vol. 3203, Lecture Notes. in Computer Science, pp. 84–94.

[8] S.B.Ors, F.Gurkaynak, E. Oswald, B. Preneel. Power-Analysis Attack on an ASIC AES implementation. In the proceedings of ITCC 2004, Las Vegas, April 5-7 2004.

[9] Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In Cryptographic Hardware and Embedded Systems–CHES 2004, 6th International Workshop, Cambridge, MA, USA, August 11-13, 2004, Proceedings, volume 3156 of Lecture Notes in Computer Science, pages 1–15. Springer, 2004.

[10] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. Proceedings of CHES 2002, volume 2535 of LNCS, pages 13-28. Springer, 2003.

[11] Http://www.opencores.org.

[12] J. Wolkerstorfer, E. Oswald, and M. Lamberger, An ASIC Implementation of the AES S-boxes, The Cryptographer's Track at the RSA Conference, CT-RSA 2002, LNCS 2271, pp. 67-78, 2002.

