Can knowledge regarding the presence of countermeasures against fault attacks simplify power attacks on cryptographic devices?

Francesco Regazzoni [1], Thomas Eisenbarth [2], Luca Breveglieri [3], Paolo Ienne [4], and Israel Koren [5]

[1] ALaRI - University of Lugano, Lugano, Switzerland. Email: regazzoni@alari.ch

[2] Horst Görtz Institute for IT Security, RUB, Bochum, Germany. Email: {eisenbarth,cpaar}@crypto.rub.de

[3] DEI - Politecnico di Milano, Milano, Italy. Email: Luca.Breveglieri@elet.polimi.it

[4] I & C - EPFL, Lausanne, Switzerland. Email: Paolo.Ienne@epfl.ch

[5] University of Massachusetts, Amherst, MA, USA. Email: koren@ecs.umass.edu

Abstract

Side-channel attacks are nowadays a serious concern when implementing cryptographic algorithms. Powerful ways for gaining information about the secret key as well as various countermeasures against such attacks have been recently developed. Although it is well known that such attacks can exploit information leaked from different sources, most prior works have only addressed the problem of protecting a cryptographic device against a single type of attack. Consequently, there is very little knowledge on how a scheme for protecting a device against one type of side-channel attack may affect its vulnerability to other types of side-channel attacks. In this paper we focus on devices that include protection against fault injection attacks (using different error detection schemes) and explore whether the presence of such fault detection circuits affects the resistance against attacks based on power analysis. Using the AES S-Box as an example, we performed attacks on the unprotected implementation as well as modified implementations with parity check circuits or residue check circuits (mod3 and mod7). In particular, we focus on the question whether the knowledge of the presence of error detection circuitry in the cryptographic device can help an attacker who attempts to mount a power attack on the device. Our results show that the presence of error detection circuitry helps the attacker even if he is unaware of this circuitry, and that the benefit to the attacker increases with the number of check bits used for the purpose of error detection.

1. Introduction

Security plays a fundamental role in today's world: the rapid growth of embedded devices executing security-sensitive applications, and the global interest in doing on-line business pose new concerns for system designers. Unfortunately, as has become evident in recent years, the use of strong cryptographic algorithms can not guarantee a sufficient level of privacy and security. In fact, increasingly simpler and cheaper attacks on cryptographic algorithms are being developed. Unlike mathematical approaches, the so called *side channel attacks* exploit the weaknesses of the hardware and/or software platform on which the algorithm is implemented in order to acquire sensitive information, rather than attempting a direct attack on the algorithm. One of the most successful examples of such attacks is that of power analysis that exploits the correlation between the power consumed by a device and the data being processed. The effectiveness of such attacks is very high

1550-5774/08 $25.00 © 2008 IEEE
DOI 10.1109/DFT.2008.53

because they do not require any particular knowledge about the implementation of the device. Besides power attacks, other side-channel attacks have been developed and shown to be very effective; for example, an attacker can get access to sensitive information by maliciously injecting faults and analyzing the faulty behavior of the system.

These unconventional forms of attack have been studied in the past and some solutions to counteract them have been proposed [2, 9, 16, 17, 18, 22, 23]. Still, there is currently no perfect protection against power attacks. However it is, possible to make the task of the attacker more difficult and more time consuming by applying several countermeasures at different levels. Similarly, fault injection attacks can be protected against using, for example, robust error detection schemes.

The focus of most previous work has been on a single type of attack, and as a result, it is not clear whether and how a countermeasure that defeats one particular attack affects the robustness against a different attack.

In this paper we concentrate on devices that are protected against fault injection attacks using different error detection schemes, and our goal is twofold: investigate whether one of the circuits is easier to attack than the others, and find out whether knowledge about the presence of an error detection circuit can be exploited by the attacker. For our study we use the AES algorithm as an example and we consider hardware implementations of the non linear transformation (Sbox) within AES. We have added error detection circuits based on parity checks as well as residue checks modulo 3 and 7, to the original implementation, and we attacked them using the *Correlation Power Attack* [6]. To compare the implementations we analyzed simulated data as well as current consumption traces obtained from transistor level simulation. The simulations, used with different attack hypothesis, allowed us to evaluate the impact of the known presence of error detection circuitry on the effectiveness of such power attacks.

The rest of the paper is organized as follows. Section 2 summarizes previous research efforts involving fault injection and power analysis attacks. Section 3 introduces the cryptographic algorithm we used as a case study, the AES, and describes our circuits for error detection. The simulation environment as well as the results are presented in Section 4. Section 5 concludes the paper.

2. Related work

Since the introduction of side-channel attacks by *Kocher et al.* [12], a large number of publications have addressed this problem. This is because such attacks – that target the device that executes the algorithm rather than the mathematical structure of the algorithm – are very powerful and often reasonably cheap. Since the attacker usually needs physical access to the device, the security threats are most severe for secure embedded system designs, in particular, for smart cards. Among the so called *side-channels* attacks, time, power, electromagnetic emanation and the deliberate injection of faults are of particular relevance [1, 11, 12].

The problem was addressed by a large number of previous works, where the common approach for defeating power analysis attacks has been to remove as much as possible the correlation between the power consumption and the data being processed, by using a combination of *hiding* and *masking* [15]. The proposed countermeasures act at different levels of the design process, ranging from algorithmic techniques [9, 23], through architectural approaches [16, 17] down to hardware-level methods [18, 22]. Still, despite the substantial amount of research, a perfect protection against such attacks is not yet available.

Defeating fault injection attacks [3, 4] is, in comparison, a simpler task since robust and efficient protection schemes can be developed based on error detection codes which have been traditionally

used in data transmission, especially for dealing with noisy channels. Parity check is an example of a classical code that was adapted to the needs of cryptographic devices. New solutions tailored to the specific needs of cryptography have also been developed. In this case, the error protection is mainly based on *Concurrent Error Detection (CED)* techniques. Typically, every time an error is detected, the detection circuit stops the normal execution of the algorithm to prevent the generation of the wrong output. As a result, the attacker is unable to view and analyze the faulty output.

Clearly, the correctness of the output can be verified by duplication of the computation either in area or in the time domain. However, both of these methods are expensive since they either double the execution time or the area requirements.

In addition to the above mentioned general approaches, some publications have focused on particular cryptographic algorithms or on particular classes of algorithms. Wolter et. al. [24] presented an implementation of the IDEA algorithm in which data are first encrypted and then, as a check, decrypted with the result compared to the original plaintext. Public key algorithms are analyzed in [8], where the authors propose an approach for providing error detection and correction by means of redundant arithmetic based on finite rings. Although comprehensive, the proposed implementation is complex and results in a higher area overhead compared to other methods.

In their work [10], Karri et al. propose a CED that is tailored to substitution-permutation network ciphers and compares the modified parity of the input with the parity of the output.

A residue-based error detection scheme for RSA was proposed in [5]. Though there remains a small possibility of undetected errors, the area overhead of the proposed scheme is very small.

The CED scheme proposed for AES in [2] uses one parity bit for every internal state byte of the AES. This scheme, which requires a limited amount of area for its implementation, detects all odd errors and in many cases even errors as well.

The main limitation of all the previous research efforts is that they target only one specific attack. Instead, while designing a scheme for protection against a given attack, it is crucial to also take into account how the implemented countermeasure would affect other possible attacks. This problem was addressed only in very few previous publications. Maingot *et al.* [13] have analyzed the impact of four different differential fault analysis countermeasures on the power analysis resistance. Their study, that was carried on using gate level simulation, shows that the power analysis vulnerability depends on the particular error detection code used.

In [14], the authors compare different error detection codes in order to provide hints about the best code selection for secure chips. The authors show that a complementary parity scheme that can improve the circuit robustness, induces higher overhead.

Transistor level simulation was performed in [20], where the authors analyze an error detection code based on parity check and discuss how this protection could affect the resistance against power based attacks and the role played by measurement noise. This paper extends the work reported in [20] by analyzing the impact of other error detection schemes (mod 3 and mod 7 residue checks) to find out which scheme is less vulnerable to power attacks. Furthermore, in this paper we study the benefits that attackers may enjoy if they are aware of the presence of error detection circuits in the cryptographic device.

3. AES and error detection circuits overview

The AES (Rijndael) [7] algorithm implements a block cipher for symmetric key cryptography. The block size is 128 bits, while the key size is 128, 192 or 256 bits. During the encryption process, four different transformations are iterated a number of times depending on the key size.

The four basic transformations are: *ShiftRows*, *SubBytes* (using SBoxes), *MixColumns*, and finally *AddRoundKey*. The added key is different in each round and these round keys are generated by a *key schedule* routine that takes the secret key and executes an expansion as specified in the standard. The same basic transformations are used during decryption, but they are applied in reverse order.

For the AES S-box, we implemented four versions of the non-linear function. The first circuit implements the non linear transformation as described in the standard, while in all the other three we added logic to provide error detection. We considered three types of error detection circuits: *parity based* and *residue code modulo 3* and *modulo 7*.

The parity check we used is the one proposed in [2]: a single even parity bit is added to every byte. The number of additional bits required for error detection based on residue code depends on the particular modulus used.

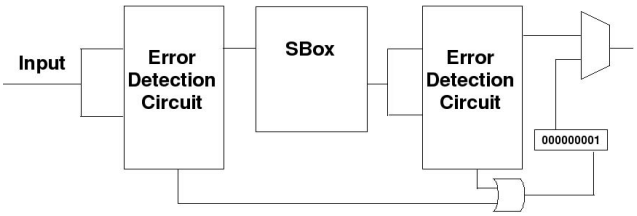

Figure 1. Block diagram of the Sbox with an attached generic error detection circuit.

Figure 1 shows an AES S-box with an added error detection circuit. The error detection circuit checks the correctness of the input and the output of the S-box. When new data enters the S-box, the check bits are separated from the data bits and an error detection is performed. If no error is detected, the 8 data bits enter the S-box circuit. The S-box produces then the result of the non-linear transformation plus the corresponding check bits. At this point the second check is performed, again as described before. If no error is detected in both checks, the output of the S-box can be forwarded to the next round transformation, otherwise, a faulty output composed of all zeros except the right most bit is generated to signal the error.

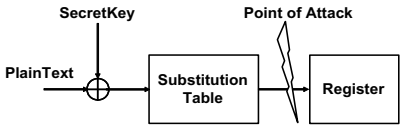

Figure 2. Our attack point: the output of the SBox.

4. Results of power attacks using simulated data

In this section we describe the circuits considered in this study and discuss the results we obtained when mounting attacks on simulated data.

We have analyzed three possible circuits for adding error detection capabilities to the AES S-Box: parity code, residue code modulo 3 and residue code modulo 7. As previously described, the goal of this paper is twofold: investigate whether one of the circuits is easier to attack than the others, and find out whether knowledge about the presence of an error detection circuit can be exploited by the attacker.

To perform such an evaluation, we consider the situation depicted in Figure 2: the input plaintext has a size of 8 bits, and only one Sbox is used following the secret key addition. In this case, our attacks target the full output of the non linear transformation. The circuit used for attacking the implementation with error detection circuits is similar to the one depicted in Figure 2, but the output of the Sbox has a number of check bits that depends on the specific scheme used: one extra bit for parity, two extra bits for residue modulo 3 and three extra bits for residue modulo 7.

To evaluate the resistance against power analysis attacks, we performed a Correlation Power Analysis (CPA) that evaluates all the key guesses using statistical correlation. In particular, the correct key guess is the one that shows the highest value for the correlation between the power consumption and the hypothesized Hamming weight.

We divided our attack into two steps. During the first one, faster but less precise, we mounted our CPA on simulated data. In the second step, we used power consumption traces collected using transistor level simulation, which are much closer to a real world attack situation. The use of simulated data has a major advantage: it is available at an early stage of the design flow and thus is of a particular interest for determining the correct point of attack and for estimating the minimum number of measurements needed to distinguish the correct key from the others.

During the first step, we obtained simulated data using the approach described in the work of Örs et al. [19] to have a first estimate of how knowledge about the error detection circuitry could help the attacker. As previously described, the target of our attack is the output of the non linear transformation that in many proposed AES architectures is stored in a register. Thus, we have developed a simulator that, using a fixed key and N random plaintexts, writes the Hamming weight of the Sbox output at each encryption cycle. We have performed this step for the normal Sbox as well as for the Sboxes with added error detection circuits. Then, using the same plaintext of the previous step, we have calculated the Hamming weight of the 8 least significant bits of the Sbox output. We have then calculated the correlation between the traces generated during step two and the ones produced during step one, increasing the number of considered plaintexts from 1 to N. In this case, the used attack hypothesis is always of 8 bit, thus this represents the situation in which the attacker is unaware of the presence of the error detection. Finally, we have calculated the Hamming weight of the Sbox output including also the check bits generated by the error detection circuit when present, and as before, we have calculated the correlation between the traces generated during step one and those produced during this step, increasing at each run the number of considered plaintexts.

The attacks we performed show that in the case of a single Sbox, the correlation for the case where the presence of the error detection circuit is known, is equal to 1, while for the case where the presence of the error detection is unknown to the attacker the correlation is lower, but the value never goes below 0.95. Based on this we can conclude that being unaware of the presence of the error detection circuits will not adversely impact the effectiveness of the power attack. We should keep in mind that in this case the values for the correlation are high because the simulations have been carried out in a noise free environment.

To have a more realistic analysis and to be able to compare different error detection circuits, we developed *VHDL* codes for all of them and performed transistor level power simulation using NANOSIM, as described in the work of Regazzoni et al. [21]. The technology library we used is the UMC 0.18μm and the options of the tool were set to provide the highest possible resolution both for time and current. As in the previous series of attacks, we randomly generated N plaintexts and using them we simulated the circuit keeping the key constant. We then added a white Gaussian noise to each trace: this noise is normally distributed with mean equal to 0 and a given variance, and mimics the typical noise generated by the measurement instruments. In particular, to simulate different noise conditions, we generated several sets of traces, obtained by adding noise with a

different variance to the same base traces generated at transistor level. We finally performed CPA calculating the correlation between the Hamming weight of the Sbox output and the power traces, increasing at each run the number of considered plaintexts.

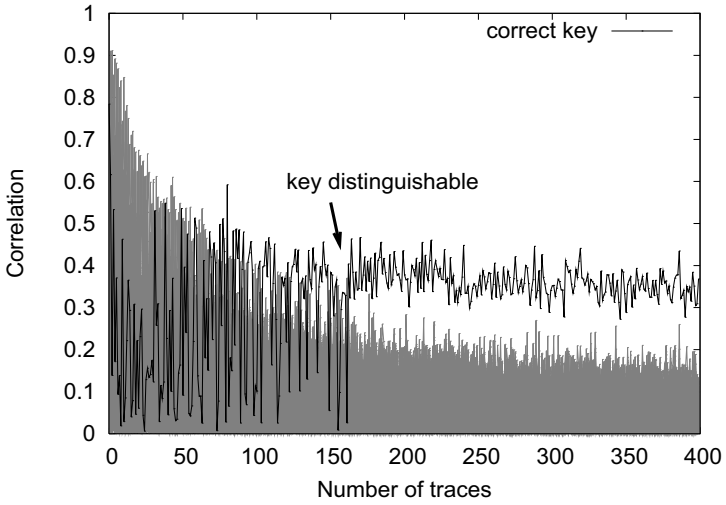

Figure 3. Attack on the AES Sbox.

Our results based on these experiments show that the presence of an error detection circuit clearly helps the attacker. These results confirm the recent observation of [20] that added parity bits have a negative impact on the Power Analysis resistance, and extend the same observation to more complex error detection schemes that have not been analyzed in the past. Furthermore, we were able to observe that there is a strong relation between the success of a power analysis attack and the number of redundancy bits used for error detection. As can be seen from Figure 3, in the case of the Sbox without error detection circuits, the correct key starts to be clearly distinguishable after 160 traces. The required number of traces is smaller when there is a parity bit (see Figure 4 where the correct key is distinguishable after 130 traces) or residue modulo 3 check bits (see Figure 5 where the correct key starts to be distinguishable after 100 traces). This indicates that the more check bits the code has, the quicker the CPA attack gets the secret key. We want to emphasize that the noise added to the traces is normally distributed, thus while considering how the error detection code helps the attacker we are not interested in checking whether we can guess correctly all the keys starting from a specific number of traces, but we are more interested in checking the trend of the time instant when the correct key becomes visible.

Additionally, to completely evaluate the impact of known presence of error detection on the success attack rate, we mounted a CPA including in the attack hypotheses the knowledge about the error detection circuit as well as without including it. Figures 5 and 6 show attacks on a Sbox that implements a residue code modulo 3, with the first figure showing the results for the case in which the presence of the detection code was known to the attacker, while the second figure presents the case when the attacker is unaware of the presence of error detection circuits. As can be seen from these figures, in both cases the correct key starts to be distinguishable soon after 100 traces and in any case it is distinguishable using fewer traces than in Figure 3, where there are no error detection circuits. Being aware of the presence of the residue code provides a small benefit to the attacker, but there is not statistically relevant evidence that the knowledge regarding the presence of a code to protect the SBOX, makes the attack significantly easier. This means, in other word, that the

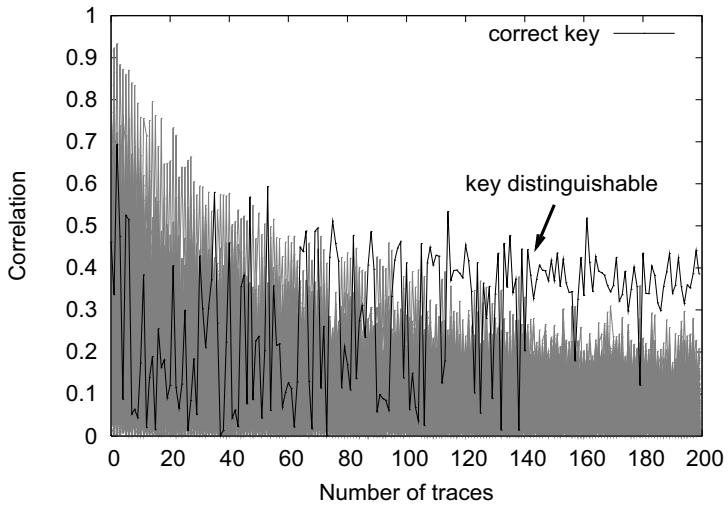

Figure 4. Attack on AES Sbox with parity based error detection.

presence of the redundancy helps the attacker even if he is not aware of it. This is due to the fact that the redundancy bits, even if not explicitly included in the hypothesis, are not random (and thus comparable to noise), but they depend on the bits targeted by the attack.

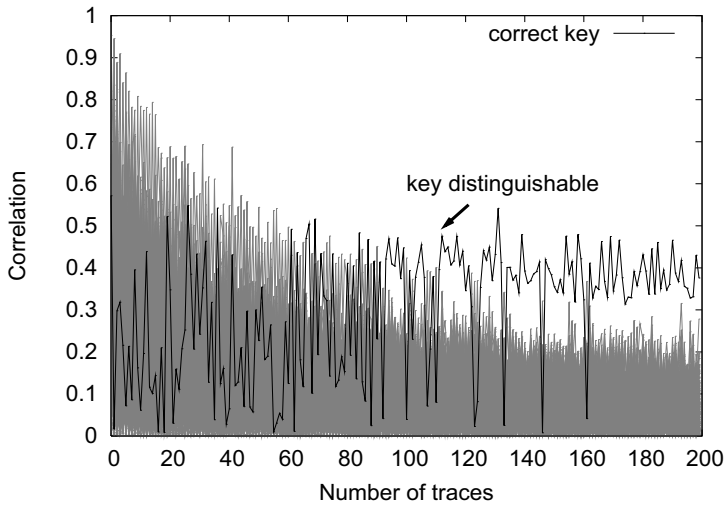

Figure 5. Attack on AES Sbox with residue code modulo 3 based error detection.

5. Conclusions

In this paper, we have evaluated the effect that different error detection circuits may have on the resistance to power analysis attacks. We focused in particular on the non linear transformation of the AES. We discussed how the attacker's knowledge of the presence of error detection circuits could affect the effectiveness of side channel attacks based on power consumption. Our results show that in a noisy situation, the check bits could help the attacker in any case, even if the presence of error

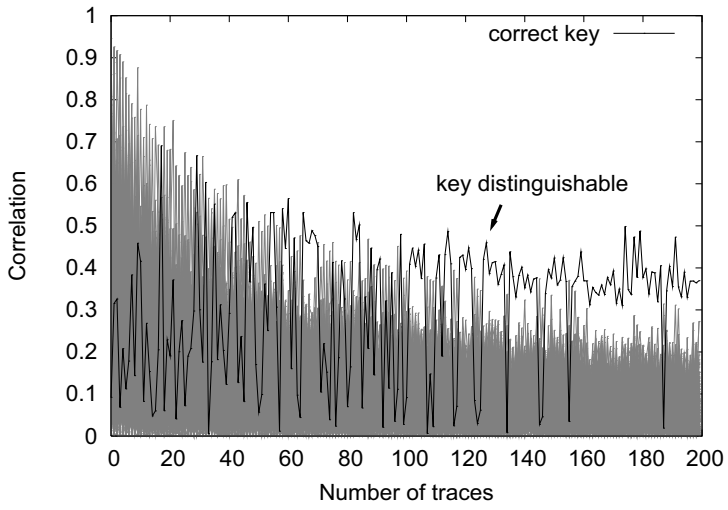

Figure 6. Attack on AES Sbox with residue code modulo 3 based error detection – the presence of the detection circuit is unknown to the attacker.

detection is unknown to the attacker. Furthermore, we showed that the higher the number of check bits is, the easier the attack is.

References

[1] F. Bao, R. H. Deng, Y. Han, A. B. Jeng, A. D. Narasimhalu, and T.-H. Ngair. Breaking public key cryptosystems on tamper resistant devices in the presence of transient faults. In *Security Protocols, 5th International Workshop*, vol. 1361 of *Lecture Notes in Computer Science*, pp. 115–124. Springer Verlag, 1998.

[2] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri. Error analysis and detection procedures for a hardware implementation of the advanced encryption standard. In *IEEE Transactions on Computers*, vol.52, pp. 492–505, 2003.

[3] E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. *Advances in Cryptology - CRYPTO*, p. 513, 1997.

[4] D. Boneh. On the Importance of Eliminating Errors in Cryptographic Computations. In *Journal of Cryptology*, vol.14, pp. 101–119. Springer, 2001.

[5] L. Breveglieri, I. Koren, P. Maistri, and M. Ravasio. Incorporating Error Detection in an RSA Architecture.

[6] E. Brier, C. Clavier, and F. Olivier. Correlation power analysis with a leakage model. In *Cryptographic Hardware and Embedded Systems — CHES 2004*, vol. 3156 of *Lecture Notes in Computer Science*, pp. 16–29. Springer, 2004.

[7] J. Daemen and V. Rijmen. AES Proposal: Rijndael. *http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf*, 1999.

[8] G. Gaubatz and B. Sunar. Robust Finite Field Arithmetic for Fault-Tolerant Public-Key Cryptography. In *2nd Workshop on Fault Diagnosis and Tolerance in Cryptography - FDTC'05*, 2005.

[9] T. Izu, B. Möller, and T. Takagi. Improved elliptic curve multiplication methods resistant against side channel attacks. In *Progress in Cryptology — INDOCRYPT 2002*, vol. 2551 of *Lecture Notes in Computer Science*, pp. 296–313. Springer Verlag, 2002.

[10] R. Karri, G. Kuznetsov, and M. Gössel. Parity-based concurrent error detection of substitution-permutation network block ciphers. In *Cryptographic Hardware and Embedded Systems — CHES 2003*, vol. 2779 of *Lecture Notes in Computer Science*, pp. 113–124. Springer, 2003.

[11] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology — CRYPTO '96*, vol. 1109 of *Lecture Notes in Computer Science*, pp. 104–113. Springer Verlag, 1996.

[12] P. C. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology — CRYPTO '99*, vol. 1666 of *Lecture Notes in Computer Science*, pp. 388–397. Springer Verlag, 1999.

[13] V. Maingot and R. Leveugle. Error Detection Code Efficiency for Secure Chips. Electronics, Circuits and Systems, ICECS '06, Dec. 2006.

[14] V. Maingot and R. Leveugle. On the use of error correcting and detecting codes in secured circuits. *Microelectronics and Electronics Conference, 2007. RME. Ph.D. Research in*, pp. 245–248, July 2007.

[15] S. Mangard, E. Oswald, and T. Popp. *Power Analysis Attacks*. Springer, 2007.

[16] D. May, H. L. Muller, and N. P. Smart. Non-deterministic processors. In *Information Security and Privacy — ACISP 2001*, vol. 2119 of *Lecture Notes in Computer Science*, pp. 115–129. Springer Verlag, 2001.

[17] D. May, H. L. Muller, and N. P. Smart. Random register renaming to foil DPA. In *Cryptographic Hardware and Embedded Systems — CHES 2001*, vol. 2162 of *Lecture Notes in Computer Science*, pp. 28–38. Springer Verlag, 2001.

[18] S. W. Moore, R. J. Anderson, P. Cunningham, R. Mullins, and G. Taylor. Improving smart card security using self-timed circuits. In *Proceedings of the 8th International Symposium on Asynchronous Circuits and Systems (ASYNC 2002)*, pp. 193–200. IEEE Computer Society Press, Apr. 2002.

[19] S. B. Örs, F. K. Gürkaynak, E. Oswald, and B. Preneel. Power-analysis attack on an asic AES implementation. In *ITCC (2)*, pp. 546–552. IEEE Computer Society, 2004.

[20] F. Regazzoni, T. Eisenbarth, J. Großschädl, L. Breveglieri, P. Ienne, I. Koren, and C. Paar. Power Attacks Resistance of Cryptographic S-boxes with added Error Detection Circuits. In *Proceedings of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'07)*, 2007.

[21] F. Regazzoni, S. Badel, T. Eisenbarth, J. Großschädl, A. Poschmann, Z. Toprak, M. Macchetti, L. Pozzi, C. Paar, Y. Leblebici, and P. Ienne. A Simulation-Based Methodology for Evaluating the DPA-Resistance of Cryptographic Functional Units with Application to CMOS and MCML Technologies. In *International Symposium on Systems, Architectures, Modeling and Simulation (SAMOS VII)*, 2007.

[22] K. Tiri, M. Akmal, and I. M. Verbauwhede. A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards. In *Proceedings of the 28th European Solid-State Circuits Conference (ESSCIRC 2002)*, pp. 403–406, Sept. 2002.

[23] C. D. Walter. MIST: An efficient, randomized exponentiation algorithm for resisting power analysis. In *Topics in Cryptology — CT-RSA 2002*, vol. 2271 of *Lecture Notes in Computer Science*, pp. 53–66. Springer Verlag, 2002.

[24] S. Wolter, H. Matz, A. Schubert, and R. Laur. On the VLSI implementation of the international data encryption algorithm IDEA. In *IEEE International Symposium on Circuits and Systems, ISCAS'95*, vol.1̃, pp. 397–400, 1995.