Lightweight Cryptography and DPA Countermeasures: A Survey

Amir Moradi[1] and Axel Poschmann[2]

[1] Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany
[2] Division of Mathematical Sciences, School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
moradi@crypto.rub.de, aposchmann@ntu.edu.sg

Abstract. The dawning Ubiquitous Computing age demands a new attacker model for the myriads of pervasive computing devices used: since a potentially malicious user is in full control over the pervasive device, additionally to the cryptographic attacks the whole field of physical attacks has to be considered. Most notably are here so-called *side channel attacks*, such as *Differential Power Analysis* (DPA) attacks. At the same time, the deployment of pervasive devices is strongly cost-driven, which prohibits expensive countermeasures. In this article we survey a broad range of countermeasures and discuss their suitability for ultra-constrained devices, such as passive RFID-tags. We conclude that adiabatic logic countermeasures, such as 2N-2N2P and SAL, seem to be promising candidates, because they increase the resistance against DPA attacks while at the same time lowering the power consumption of the pervasive device.

1 Introduction

Mark Weiser's famous vision of *ubiquitous computing* (ubicomp) [47], which is widely believed to be the next paradigm in information technology, seems to become reality in the near future, since increasingly everyday items are enhanced to pervasive devices by embedding computing power. The mass deployment of pervasive devices promises on the one hand many benefits (e.g. optimized supply-chains), but on the other hand, many foreseen applications are security sensitive (military, financial or automotive applications), not to mention possible privacy issues. With the widespread presence of embedded computers in such scenarios security is a striving issue, because the potential damage of malicious attacks also increases. Even worse, pervasive devices are deployed in a hostile environment, *i.e.* an adversary has physical access to or control over the devices, which enables the whole field of physical attacks. Not only the adversary model is different for ubicomp, but also its optimisation goals are significantly different from that of traditional application scenarios: high throughput is usually not an issue but power, energy and area are sparse resources. Due to the harsh cost constraints for ubicomp applications only the least required amount of computing power will be realized. If computing power is fixed and cost are variable, Moore's Law leads to the paradox of an increasing demand for lightweight solutions.

R. Sion et al. (Eds.): FC 2010 Workshops, LNCS 6054, pp. 68–79, 2010.

In this article we are going to address the issue of lightweight side-channel countermeasures. Our main contribution is a survey of countermeasures on different architectural levels (cell, gate, algorithmic) and an evaluation of their suitability for constrained devices. Our main metrics are the area and timing overhead, but we also take practical evaluations into account to identify a set of countermeasures that seem to be promising for constrained devices.

The remainder of this article is organized as follow: In Section 2 we are going to briefly highlight the hardware properties of basic building blocks, such as Boolean operations and flipflops. Subsequently in Section 3 we introduce to Side channel attacks and several previously proposed countermeasures. Then in Section 4 we will evaluate a selection of countermeasures with regard to their suitability for constrained devices. Finally this paper is concluded in Section 5.

2 Hardware Properties of Cryptographic Building Blocks

Block ciphers take a block of data and a key as input and transform it to a ciphertext, often using a roundfunction that is iterated several times. The intermediate state is called *data state* and *key state*, respectively. While software implementations have to process single operations in a serial manner, hardware implementations offer more flexibility for parallelization and serialization. Generally speaking there exist three major architecture strategies for the implementation of block ciphers: *serialized*, *round-based*, and *parallelized.* In a *serialized* architecture only a fraction of a single round is processed in one clock cycle. These lightweight implementations allow to reduce area and power consumption at the cost of a rather long processing time. If a complete round is performed in one clock cycle, we have a *round-based* architecture. This implementation strategy usually offers the best time-area product and throughput per area ratio. A *parallelized* architecture processes more than one round per clock cycle, leading to a rather long critical path. A longer critical path leads to a lower maximum frequency but also requires the gates to drive a higher load (fanout), which results in larger gates with a higher power consumption. By inserting intermediate registers (a technique called *pipelining*), it is possible to split the critical path into fractions, thus increasing the maximum frequency. Once the pipeline is filled, a complete encryption can be performed in one clock cycle with such an architecture. Consequently, this implementation strategy yields the highest throughput at the cost of high area demands. Furthermore, since the pipeline has to be filled, each pipelining stage introduces a delay of one clock cycle.

In the context of lightweight cryptography, clearly serialized implementations are the most important architecture, since they allow to significantly reduce the area and power demands. In order to compare the area requirements independently of the technology used, it is common to state the area as *gate equivalents* [GE]. One GE is equivalent to the area which is required by the two-input NAND gate with the lowest driving strength of the appropriate technology. The area in GE is derived by dividing the area in μm^2 by the area of a two-input NAND gate. However, it is not easy to compare the power consumption of different technologies.

In order to reuse the same hardware resources in a serialized or round-based implementation, data and key state have to be stored. Since external memory is often not available for cryptographic applications or draws too much current (e.g. on passive RFID-tags), the state has to be maintained in registers using flipflops. Unfortunately flipflops have a rather large area and power demand, for example, when using the *Virtual Silicon (VST)* standard cell library based on the *UMC L180* 0.18μ *1P6M Logic process* (UMCL18G212T3, [46]), flipflops require between 5.33 GE and 12.33 GE to store a single bit (see Table 1). The gate count differs so significantly for different cells because the first cell (HDDFFPB1) consists only of a simple D flipflop itself, while the latter one (HDSDERSPB1) comprises of a multiplexer to select one of two possible inputs for storage and a D flipflop with active-low enable, asynchronous clear and set. There exists a wide variety of flipflops of different complexity between these two extremes. A good trade-off between efficiency and useful supporting logic provide the two flipflop cells HDSDEPQ1 and HDSDFPQ1. Both are scan flipflops, which means that beside the flipflop they also provide a multiplexer. The latter one is also capable of being gate clocked, which is an important feature to lower power consumption. Storage of the internal state typically accounts for at least 50 % of the total area and power consumption. E.g. the area requirements of storage logic accounts for 55 % in the case of a round-based PRESENT [3] and for 86% in the case of a serialized PRESENT [34], while for a serialized AES it accounts for 60 % of the area and half of the current consumption (*i.e.* 52 %) [7]. Therefore implementations of cryptographic algorithms for low-cost tag applications should aim to minimize the storage required.

Table 1. Area requirements and corresponding gate count of selected standard cells of the UMCL18G212T3 library [46]

Standard cell	Cell name	Area in μm^2	GE
NOT	HDINVBD1	6.451	0.67
NAND	HDNAN2D1	9.677	1
NOR	HDNOR2D1	9.677	1
AND	HDAND2D1	12.902	1.33
OR	HDOR2D1	12.902	1.33
MUX	HDMUX2D1	22.579	2.33
XOR (2-input)	HDEXOR2D1	25.805	2.67
XOR (3-input)	HDEXOR3D1	45.158	4.67
D Flip flop	HDDFFPB1	51.61	5.33
Scan D flipflop /w enable	HDSDFPQ1	58.061	6
Scan flipflop	HDSDEPQ1	83.866	8.67
complex Scan flipflop	HDSDERSPB1	119.347	12.33

The term *combinatorial elements* includes all the basic Boolean operations such as NOT, NAND, NOR, AND, OR, and XOR. It also includes some basic logic functions such as multiplexers (MUX). It is widely assumed that the gate count for these basic operations is typically independent of the library used. However, in [34] it has been shown that ASIC implementation results of a serialized PRESENT in different technologies range from $1,000$ GE to $1,169$ GE. This indicates that also the gate count for basic logic gates differs depending on the used standard-cell library. For the *Virtual Silicon (VST)* standard cell library based on the *UMC L180* 0.18μ *1P6M Logic process* (UMCL18G212T3, [46]) the figures for selected two-input gates with the lowest driving strength is given in Table 1. Note that in hardware XOR and MUX are rather expensive when compared to the other basic Boolean operations.

In the next section we will introduce background information of *Differential Power Analysis* attacks and their countermeasures.

3 Introduction to DPA and Countermeasures

Although nowadays side-channel attacks, after the first publication of power analysis attacks in [16], are known as a serious threat for devices performing cryptographic operations, in fact this kind of attacks has been accidentally discovered in 1943 [26]. These attacks exploit the fact that the execution of a cryptographic algorithm on a physical device leaks information about the processed data and/or executed operations through side channels, e.g., power consumption [16], execution time [15] and electromagnetic radiation [8]. As presented in a number of publications, side-channel attacks particularly power analysis attacks are considered as an extremely powerful and practical tool for breaking cryptographic devices.

By measuring and evaluating the power consumption of a cryptographic device, information-dependent leakage is exploited and combined with the knowledge about the plaintext or ciphertext (in contrary to mathematical cryptanalyses which require pairs of plain- and ciphertexts) in order to extract, e.g., a secret key. Since intermediate results of the computations can be derived from the leakage, e.g., from the Hamming weight of the data processed in a software implementation, a divide-and-conquer strategy becomes possible, i.e., the secret key could be recovered byte by byte.

A *Simple Power Analysis (SPA)* attack, as introduced in [16], relies on visual inspection of power traces, e.g., measured from an embedded microcontroller of a smartcard. The aim of an SPA is to reveal details about the execution of the program flow of a software implementation, like the detection of conditional branches depending on secret information. Contrary to SPA, *Differential Power Analysis (DPA)* utilizes statistical methods and evaluates several power traces with often uniformly distributed known plaintexts or known ciphertexts. A DPA requires no knowledge about the concrete implementation of the cipher and can hence be applied to any unprotected black box implementation. According to intermediate values depending on key hypotheses the traces are divided into sets or correlated to estimated power values, and then statistical tools, e.g., difference of estimated means [16], correlation coefficient [4], and estimated mutual information [10], indicate the most probable hypothesis amongst all partially guessed key hypotheses.

Several schemes have been proposed to protect cryptographic implementations against DPA attacks. A DPA countermeasure aims at preventing a dependency between the power consumption of a cryptographic device and intermediate values of the executed algorithm [17]. *Hiding* and *Masking* are amongst the most common countermeasures on either the hardware or the software level. The goal of *Hiding* methods is to increase the noise factor [48] or to equalize the power consumption values independently of the processed data while *Masking* rely on randomizing key-dependent intermediate values processed during the execution of the cipher. The most common proposed countermeasures can be classified as follows:

- **Cell Level** (DPA-resistant logic styles): Counteracting DPA attacks at the cell level means that the logic cells of a circuit are implemented in such a way that their power consumption is independent of the processed data and the performed operations [17]. During the last years, several proposals as DPA-resistant logic style have been made and a selection is listed below:
  - **Sense Amplifier Based Logic (SABL)** [42], which is a dual-rail precharge logic, is designed to have a constant internal power consumption independent of the processed logic values. In order to achieve this aim, a full-custom design tool must be used to balance all the internal capacitances of the final layout.
  - **Wave Dynamic Differential Logic (WDDL)** [43] and **Masked Dual-rail Precharge Logic (MDPL)** [32] have been designed to avoid the usage of a full-custom design tool. However, their implementations show strong data-dependent leakage [39,31,36] which makes them vulnerable to straightforward DPA attacks.
  - **Random Switching Logic (RSL)** [38,40] employs several random bits for a non-linear combinational circuit and needs a special design flow to reach the desired level of protection. For instance a practical implementation showed vulnerability to a single-bit DPA attack [35].
  - **Dual-rail Transition Logic (DTL)** [24], which aims at randomly changing the logic values and presenting the desired data at the same time, has not been practically evaluated yet and its effectiveness is still uncertain.
  - **Charge Recovery Logics** have been proposed for low-power applications, and some of them, so-called *adiabatic logic* styles, have been investigated from DPA-resistance point of view in [22] and [14]. Adiabatic logic uses a time-varying voltage source and its slopes of transition are slowed down. This reduces the energy dissipation of each transition. In short the idea of adiabatic logic is to use a trapezoidal power-clock voltage rather than fixed supply voltage. As a consequence the power consumption of a circuit is reduced while at the same time its resistance against side-channel attacks is greatly enhanced.

- **Masking:** Randomizing the values which are processed by the cryptographic device can be performed at different levels of abstraction:
  - **Gate Level:** Masking at the gate level is performed by considering a number of mask bits for each logic value of the circuit. There are a number of proposals on how to use mask bits at the gate level, e.g., [12], [44] and [45]. However, practical realization of such schemes faces with glitches which inherently happen on logic circuit and cause vulnerability to DPA attacks [18].
  - **Algorithm Level:** According to the masking scheme, e.g., additive or multiplicative, non-linear functions of the given cipher must be redesigned to fulfill the desired level of security. There are a set of publications on contributing a masking scheme on the AES substitution function, e.g., [29] and [2]. Nevertheless, their practical investigations show vulnerability to those DPA attacks which consider glitches of the combinational circuit as the hypothetical power model [19]. Moreover, there are some proposals which are provably secure, e.g., [49] and [5]. Though they have not been practically investigated, the same vulnerability to glitches is expected.

    A threshold implementation of Sboxes has been proposed in [27,28] to avoid the effect of glitches, but it has not been practically verified yet.
- **Hiding:** Randomizing the amounts of power consumption in order to hide the sensitive operation is often performed on software implementations by shuffling the execution of operations and/or by insertion of dummy operations [17]. Although this class of countermeasures can not perfectly protect against DPA attacks, its combination with algorithmic masking, which has been introduced in [11], provides a reasonable level of protection [41].

  Randomly permuting intermediate values using permutation tables [13] also can be considered as a hiding scheme, but its efficiency has been investigated as a vulnerability has bee reported in [33]. Moreover, dynamic reconfiguration, which has been proposed in [20], can be considered as a realization of shuffling in hardware.

4 Comparison of Countermeasures

In this section we will evaluate the countermeasures introduced in the previous section with regard to the following criteria:

Area Overhead: The area overhead of every countermeasure is clearly one of the most important metrics, when low-cost devices are considered, since the cost of an ASIC are proportional to its area. These figures are either obtained from the corresponding publications or estimated. Therefore they should primarily not be seen as precise figures, but rather as an indicator in what range a countermeasures is to be expected to increase the area.

Timing Overhead: Typically timing is not critical in many low-cost applications as only rather small amounts of data are going to be processed. However, the energy consumption is directly proportional to the amount of clock cycles required. Therefore the timing overhead is an important measure for active (i.e. battery powered) constrained devices, rather than for passive (i.e. without an own power supply) constrained devices. Similar to the area overhead these figures are either obtained from the corresponding publications or are estimated and should be viewed as rough guidelines rather than precise figures.

Practical Evaluation: It has turned out that countermeasures that have been shown to be provably secure by using simulated power consumption can be attacked when real ASIC implementations are used, e.g., [29] vs. [19]. On the other hand, theoretical attacks on simulated power consumptions have been shown to be impractical on real world ASIC implementations, e.g., [32] vs. [31]. Therefore practical evaluation of a countermeasure is crucial for a more precise evaluation of the security level that can be achieved with this countermeasure. Furthermore, this column is a good indicator for future work as it shows where prototyping of an ASIC has been done already.

Known Leakages: This column lists publications that have found theoretical or practical leakages of the countermeasure.

In the following some notes on Table 2, which summarizes a comparison between the most promising countermeasures, are given. MDPL [32] has only around halve the speed, because MDPL gates consist of two P-N networks due to the usage of majority gates, i.e., a basic majority cell followed by an inverter. Area overhead ranges from 2 for a buffer, over 3.5 for a D-type flipflop and up to 6 for an XNOR gate. A prototyped ASIC implementation of the AES resulted in an area overhead factor of around 5, a power overhead factor of 11 and a timing overhead factor of 2.6 [30]. Several leakages have been found for MDPL [37,9,23,21,36] and a chip has been prototyped and evaluated by the authors of MDPL in [31]. Finally, the authors have proposed an improved MDPL, called iMDPL [31]. However, iMDPL requires 3 times more area than MDPL, thus increasing the total area overhead factor to around 15, *i.e.* an implementation in iMDPL is around 15 *times* larger than a plain CMOS implementation. Furthermore, the leakages reported in [36,9,21] also hold for iMDPL.

RSL [40,38] doubles the area requirements while halving the speed for the maximum frequency, since timing is not critical, there can no delay be expected in low frequency typical for low-cost devices. However, after prototyping an ASIC a leakage has been reported in [35].

Charge recovery logics, e.g., 2N-2N2P [22] and SAL [14], increase the area by a factor between 2 and 4. However, the power consumption is *less* than for standard CMOS circuits. Since their DPA-resistance increases with lower frequencies, it makes them particular valuable for low-power low throughput applications, such as passive RFID-tags. No charge recovery logic has been yet practically evaluated and no leakages have been fund so far. It seems to be one of the most promising candidates for future evaluation. However, since it is a full-custom design no standard-cell design flow can be used.

All gate-level masking schemes [12,44,45] have been shown to be susceptible in the presence of glitches [18] and thus are not considered any further by us.

Moreover, both algorithmic masking approaches [2] and [29] are susceptible to toggle count attackes as shown in [19].

Canright algorithmic masking [5] yields a very compact S-box of the AES that is 2.7 times as large as an unprotected S-box for the first round and 2.2 times larger for every subsequent round. A masked AES implementation would require to also store the mask bits which would double the area requirements for storage. All together the area overhead factor is estimated to be 2.5. Since it has not yet practically evaluated it seems to be an interesting candidate for further investigations, especially its resistance to glitching attacks. Zakeri algorithmic masking [49] also increases the area by a factor of around 4, which is rather large. However, there has been no practical evaluation so far and no leakage has been found.

Nikova algorithmic masking based on secret sharing [27,28] has not been practically evaluated so far. It requires to store at least two additional mask bits for every masked bit. Given the fact that especially in lightweight implementations storage accounts for the majority of the gate count, it is fair to estimate the hardware overhead with a factor of 3. However, this countermeasures has not

Table 2. Area and Timing overhead of several side channel countermeasures(1) (estimated values are denoted by *)

Countermeasure			Overhead factor		Pract.	Leakage
Level	Type/Name	Ref.	Area	Time	eval.	found in
Cell	MDPL	[32]	5	2.6	yes	[9,23,25,30,31,36,37]
	iMDPL	[31]	*15	*6	no	[9,21,36]
	RSL	[40,38]	2	2	yes	[35]
	DTL	[24]	*11	*4	no	none
	2N-2N2P	[22]	*2	(2)	no	none
	SAL	[14]	*4	(2)	no	none
Gate	Private Circuits	[12]	(3)	(3)	no	[18]
	Masking	[44,45]	*10	*5	no	[18]
Alg.	Masking	[2]	*8	*5	no	[19]
	Masking	[29]	*6	*4	no	[19]
	Masking	[5]	2.5	3	no	none
	Masking	[49]	4	3	no	none
	Secret Sharing	[27,28]	*3	*1.3	no	none
	Shuffling + Masking	[11]	7	10	yes	[41]
	Rand. Perm. Tab.	[13]	2.5	12	yes	[33]
	Dyn. Reconf.	[20]	4.75	3.36	yes	none

(1) Note that the overheads vary by different algorithms and architectures. The values presented in this table are mostly based on implementations of the AES encryption algorithm, and we did our best to consider the same architecture for all countermeasures.

(2) suitable for low-throughput applications.

(3) depends on the level of protection, e.g., area overhead would be an order of $O(nt^2)$ where n is the size of the original circuit and t is related to the desired protection level.

been practically evaluated and seems to be an interesting candidate for future investigations.

Dynamic reconfiguration [20] increases the area requirements by a factor of 4.75 and reduces the maximum clock frequency by a factor of 3.36. However, since lightweight applications typically do not need high throughput the timing overhead is not important, but the area overhead is already rather high.

5 Conclusions

The structural problem of most of todays SCA countermeasures is that they significantly increase the area, timing and power consumption of the implemented algorithm compared to an unprotected implementation. Furthermore, many countermeasures require random numbers, hence also a TRNG or a PRNG[1] has to be available. Since this will also increase the cost of an implementation of the algorithm, it will delay the break-even point and hence the mass deployment of some applications. For ultra-constrained applications, such as passive RFID tags, some countermeasures pose an impregnable barrier, because the power consumption of the protected implementation is much higher than what is available.

Power optimization techniques are an important tool for lightweight implementations of specific pervasive applications and might ease the aforementioned problem. On the one hand they also strengthen implementations against side channel attacks, because they lower the power consumption (the signal), which decreases the signal to noise ratio (SNR). However, on the other hand power saving techniques also *weaken* the resistance against side channel attacks. One consequence of the power minimization goal is that in the optimal case only those parts of the data path are active that process the relevant information. Furthermore, the width of the data path, *i.e.* the amount of bits that are processed at one point in time, is reduced by serialization. This however implies that the algorithmic noise is reduced to a minimum, which reduces the amount of required power traces for a successful side channel attack. Even worse, the serialized architecture allows the adversary a divide-and-conquer approach which further reduces the complexity of a side channel attack. Summarizing, it can be concluded that lightweight implementations greatly enhance the success probability of a side channel attack. The practical side channel attack [6] on *KeeLoq* applications [1] impressively underline this conclusions.

Adiabatic logics, like other DPA countermeasures, have an area overhead, but decrease the (instantaneous) power consumption by decreasing the frequency. As a consequence the resistance of the corresponding circuit against side-channel attacks is extremely increased. Especially for pervasive devices adiabatic logic styles seem to be a promising SCA countermeasure and practical evaluations of these logic styles will be worth reading. Furthermore, also the approach taken by Nikova *et al.* [27,28] is a promising candidate, because it has a moderate area overhead and was theoretically proven to be secure against DPA attacks.

[1] True Random Number Generator, Pseudo Random Number Generator.

Acknowledgment

The research was supported in part by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03.

References

1. Keeloq Algorithm (November 2006), http://en.wikipedia.org/wiki/KeeLoq
2. Akkar, M., Giraud, C.: An Implementation of DES and AES Secure against Some Attacks. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 309–318. Springer, Heidelberg (2001)
3. Bogdanov, A., Leander, G., Knudsen, L., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT - An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
4. Brier, E., Clavier, C., Olivier, F.: Correlation Power Analysis with a Leakage Model. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 16–29. Springer, Heidelberg (2004)
5. Canright, D., Batina, L.: A Very Compact "Perfectly Masked" S-Box for AES. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 446–459. Springer, Heidelberg (2008)
6. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., Shalmani, M.T.M.: On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoqCode Hopping Scheme. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 203–220. Springer, Heidelberg (2008)
7. Feldhofer, M., Wolkerstorfer, J., Rijmen, V.: AES Implementation on a Grain of Sand. IEE Proceedings on Information Security 152(1), 13–20 (2005)
8. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic Analysis: Concrete Results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (2001)
9. Gierlichs, B.: DPA-Resistance Without Routing Constraints? – A Cautionary Note About MDPL Security. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 107–120. Springer, Heidelberg (2007)
10. Gierlichs, B., Batina, L., Tuyls, P., Preneel, B.: Mutual Information Analysis. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 426–442. Springer, Heidelberg (2008)
11. Herbst, C., Oswald, E., Mangard, S.: An AES Smart Card Implementation Resistant to Power Analysis Attacks. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 239–252. Springer, Heidelberg (2006)
12. Ishai, Y., Sahai, A., Wagner, D.: Private Circuits: Securing Hardware against Probing Attacks. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 463–481. Springer, Heidelberg (2003)
13. Coron, J.-S.: A New DPA Countermeasure Based on Permutation Tables. In: Ostrovsky, R., De Prisco, R., Visconti, I. (eds.) SCN 2008. LNCS, vol. 5229, pp. 278–292. Springer, Heidelberg (2008)
14. Khatir, M., Moradi, A., Ejlali, A., Shalmani, M.T.M., Salmasizadeh, M.: A Secure and Low-Energy Logic Style using Charge Recovery Approach. In: SLPED 2008, pp. 259–264. ACM, New York (2008)
15. Kocher, P.C.: Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996)

16. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
17. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. Springer, Heidelberg (2007)
18. Mangard, S., Popp, T., Gammel, B.M.: Side-Channel Leakage of Masked CMOS Gates. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 351–365. Springer, Heidelberg (2005)
19. Mangard, S., Pramstaller, N., Oswald, E.: Successfully Attacking Masked AES Hardware Implementations. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 157–171. Springer, Heidelberg (2005)
20. Mentens, N., Gierlichs, B., Verbauwhede, I.: Power and Fault Analysis Resistance in Hardware through Dynamic Reconfiguration. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 346–362. Springer, Heidelberg (2008)
21. Moradi, A., Eisenbarth, T., Poschmann, A., Rolfes, C., Paar, C., Shalmani, M.T.M., Salmasizadeh, M.: Information Leakage of Flip-Flops in DPA-Resistant Logic Styles. Cryptology ePrint Archive, Report 2008/188 (2008), `http://eprint.iacr.org/`
22. Moradi, A., Khatir, M., Salmasizadeh, M., Shalmani, M.M.: Charge Recovery Logic as a Side Channel Attack Countermeasure. In: ISQED 2009, pp. 686–691 (2009)
23. Moradi, A., Salmasizadeh, M., Shalmani, M.T.M.: Power Analysis Attacks on MDPL and DRSL Implementations. In: Nam, K.-H., Rhee, G. (eds.) ICISC 2007. LNCS, vol. 4817, pp. 259–272. Springer, Heidelberg (2007)
24. Moradi, A., Shalmani, M.T.M., Salmasizadeh, M.: Dual-Rail Transition Logic: A Logic Style for Counteracting Power Analysis Attacks. Computers and Electrical Engineering 35(2), 359–369 (2009)
25. Mulder, E.D., Gierlichs, B., Preneel, B., Verbauwhede, I.: Practical DPA Attacks on MDPL. Cryptology ePrint Archive, Report 2009/231 (2009), `http://eprint.iacr.org/`
26. National Security Agency: TEMPEST: A Signal Problem. Cryptologic Spectrum 2(3) (1972) (declassified 2007)
27. Nikova, S., Rechberger, C., Rijmen, V.: Threshold Implementations Against Side-Channel Attacks and Glitches. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 529–545. Springer, Heidelberg (2006)
28. Nikova, S., Rijmen, V., Schläffer, M.: Secure Hardware Implementations of Non-Linear Functions in the Presence of Glitches. In: Lee, P.J., Cheon, J.H. (eds.) ICISC 2008. LNCS, vol. 5461, pp. 218–234. Springer, Heidelberg (2009)
29. Oswald, E., Mangard, S., Pramstaller, N., Rijmen, V.: A Side-Channel Analysis Resistant Description of the AES S-box. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 413–423. Springer, Heidelberg (2005)
30. Popp, T., Kirschbaum, M., Mangard, S.: Practical Attacks on Masked Hardware. In: Fischlin, M. (ed.) RSA Conference 2009. LNCS, vol. 5473, pp. 211–225. Springer, Heidelberg (2009)
31. Popp, T., Kirschbaum, M., Zefferer, T., Mangard, S.: Evaluation of the Masked Logic Style MDPL on a Prototype Chip. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 81–94. Springer, Heidelberg (2007)
32. Popp, T., Mangard, S.: Masked Dual-Rail Pre-charge Logic: DPA-Resistance without Routing Constraints. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 172–186. Springer, Heidelberg (2005)
33. Prouff, E., McEvoy, R.: First-Order Side-Channel Attacks on the Permutation Tables Countermeasure. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 81–96. Springer, Heidelberg (2009)

34. Rolfes, C., Poschmann, A., Leander, G., Paar, C.: Ultra-Lightweight Implementations for Smart Devices - Security for 1000 Gate Equivalents. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 89–103. Springer, Heidelberg (2008)
35. Saeki, M., Suzuki, D., Shimizu, K., Satoh, A.: A Design Methodology for a DPA-Resistant Cryptographic LSI with RSL Techniques. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 189–204. Springer, Heidelberg (2009)
36. Schaumont, P., Tiri, K.: Masking and Dual-Rail Logic Don't Add Up. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 95–106. Springer, Heidelberg (2007)
37. Suzuki, D., Saeki, M.: Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 255–269. Springer, Heidelberg (2006)
38. Suzuki, D., Saeki, M., Ichikawa, T.: Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive, Report 2004/346 (2004), `http://eprint.iacr.org/`
39. Suzuki, D., Saeki, M., Ichikawa, T.: DPA Leakage Models for CMOS Logic Circuits. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 366–382. Springer, Heidelberg (2005)
40. Suzuki, D., Saeki, M., Ichikawa, T.: Random Switching Logic: A New Countermeasure against DPA and Second-Order DPA at the Logic Level. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E90-A(1), 160–168 (2007)
41. Tillich, S., Herbst, C.: Attacking State-of-the-Art Software Countermeasures - A Case Study for AES. In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 228–243. Springer, Heidelberg (2008)
42. Tiri, K., Akmal, M., Verbauwhede, I.: A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. In: ESSCIRC 2002, pp. 403–406 (2002)
43. Tiri, K., Verbauwhede, I.: A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In: DATE 2004, pp. 246–251 (2004)
44. Trichina, E.: Combinational Logic Design for AES Subbyte Transformation on Masked Data,
`http://eprint.iacr.org/2003/236`
45. Trichina, E., Korkishko, T., Lee, K.H.: Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results. In: Dobbertin, H., Rijmen, V., Sowa, A. (eds.) AES 2005. LNCS, vol. 3373, pp. 113–127. Springer, Heidelberg (2005)
46. Virtual Silicon Inc. 0.18 μm VIP Standard Cell Library Tape Out Ready, Part Number: UMCL18G212T3, Process: UMC Logic 0.18 μm Generic II Technology: 0.18μm (July 2004)
47. Weiser, M.: The Computer for the 21st Century. ACM SIGMOBILE Mobile Computing and Communications Review 3(3), 3–11 (1999)
48. Yang, S., Wolf, W., Vijaykrishnan, N., Serpanos, D.N., Xie, Y.: Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach. In: DATE 2005, pp. 64–69. IEEE Computer Society, Los Alamitos (2005)
49. Zakeri, B., Salmasizadeh, M., Moradi, A., Tabandeh, M., Shalmani, M.: Compact and Secure Design of Masked AES S-Box. In: Qing, S., Imai, H., Wang, G. (eds.) ICICS 2007. LNCS, vol. 4861, pp. 216–229. Springer, Heidelberg (2007)