Towards Fresh Re-Keying with Leakage-Resilient PRFs: Cipher Design Principles and Analysis

Sonia Belaïd[1], Fabrizio De Santis[2,3], Johann Heyszl[4], Stefan Mangard[3], Marcel Medwed[5], Jörn-Marc Schmidt[6], François-Xavier Standaert[7], Stefan Tillich[8]

[1] Ecole Normale Supérieure and Thales Communications, France.
[2] Institute for Security in Information Technologies, Technical University of Munich.
[3] Infineon Technologies AG, Neubiberg, Germany.
[4] Fraunhofer Research Institution AISEC, Munich, Germany.
[5] NXP Semiconductors, Gratkorn, Austria.
[6] IAIK, Graz University of Technology, Austria.
[7] ICTEAM/ELEN/Crypto Group, Université catholique de Louvain, Belgium.
[8] Department of Computer Science, University of Bristol, UK.

Abstract. Leakage-resilient cryptography aims at developing new algorithms for which physical security against side-channel attacks can be formally analyzed. Following the work of Dziembowski and Pietrzak at FOCS 2008, several symmetric cryptographic primitives have been investigated in this setting. Most of them can be instantiated with a block cipher as underlying component. Such an approach naturally raises the question whether certain block ciphers are better suited for this purpose. In order to answer this question, we consider a leakage-resilient re-keying function, and evaluate its security at different abstraction levels. That is, we study possible attacks exploiting specific features of the algorithmic description, hardware architecture and physical implementation of this construction. These evaluations lead to two main outcomes. First, we complement previous works on leakage-resilient cryptography and further specify the conditions under which they actually provide physical security. Second, we take advantage of our analysis to extract new design principles for block ciphers to be used in leakage-resilient primitives. While our investigations focus on side-channel attacks in the first place, we hope these new design principles will trigger the interest of symmetric cryptographers to design new block ciphers combining good properties for secure implementations and security against black box (mathematical) cryptanalysis.

1 Introduction

Securing embedded devices against side-channel attacks is an important challenge in modern cryptography. Because of their technology-dependent nature, protections against these attacks usually require combining ideas at different abstraction levels, e.g. exploiting noise in physical processes and randomness in hardware/software designs [22]. In the context of symmetric cryptography, a recent and concurrent trend has investigated the opportunities to analyze new primitives, better suited for physically-secure implementations. Dziembowski and Pietrzak's leakage-resilient cryptography is one of the most investigated models for this purpose [7], and several proposals of pseudorandom generators (PRGs)/stream ciphers, pseudorandom functions (PRFs) and pseudorandom permutations (PRPs) have been considered in this setting [6,9,27,35,39,40]. These new constructions naturally raise interesting open questions regarding the practical relevance of formal models for physical security analysis. Yet, they are all based on some kind of re-keying strategies (i.e. reminiscent from Kocher's early patents [17]). Hence, and somewhat independent of these

modeling issues, it may very well be that (small variations of) ideas proposed in such theoretical works actually provide significantly enhanced security against large categories of "practical attacks". Since another possible drawback of leakage-resilient cryptography is its significant performance overheads, it naturally suggests an intermediate line of research, where the security of leakage-resilient primitives is analyzed in front of actual side-channel adversaries, in order to mitigate these overheads. This approach has been recently followed by Medwed et al. for the case of leakage-resilient PRFs [26].

In this paper, we embrace a similar strategy and further study the possibilities to design secure and efficient leakage-resilient PRFs. In particular, we focus on their instantiation using block ciphers, which is motivated by the large literature on side-channel attacks and countermeasures for this type of building blocks. In this context, our main goal is to investigate new design principles that would be best suited for the secure implementation of such primitives. For this purpose and as a starting point, we analyze the physical security of a generic block cipher construction, aimed to be used in the re-keying scheme represented in Figure 1. This re-keying essentially uses a function g to re-key a block cipher f with a master key k and a public random nonce r. For each block of message, a fresh key is computed as $k^\star = g_k(r)$, and then used to generate the ciphertext $c = f_{k^\star}(m)$. One important advantage of this scheme (put forward and analyzed in [25]) is that (informally): (i) from the mathematical point of view, f has to be cryptographically strong while g only requires some minimum diffusion properties, (ii) by contrast, from the implementation point of view f only needs to be secure against Simple Power Analysis (SPA)[1], while g has to resist both SPA and Differential Power Analysis (DPA)[2]. The solution previously proposed in [25] was to use a modular multiplication for g, which benefits from the feature of being easy to mask [5,11]. Yet, and despite being promising from a security and performance point of view, this proposal is quite specific to one countermeasure (namely, masking) that has proved to be quite effective in software [32], but may turn out to be difficult to implement in hardware [23]. As a result, we propose to investigate alternative candidates for the g function, and focus on hardware implementation issues, in order to increase the versatility of the design space for fresh re-keying.

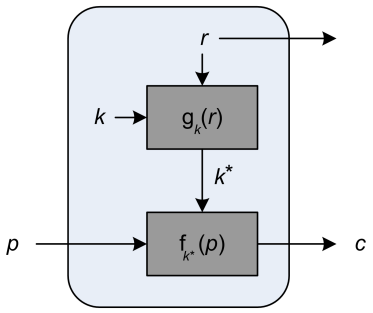

Fig. 1. Fresh re-keying: basic principle.

[1] i.e. side-channel attacks with data complexity 1, essentially.
[2] i.e. side-channel attacks with larger data complexity, essentially.

Our contributions. The CHES 2012 work of Medwed et al. is based on a new assumption that identical components (e.g. S-boxes) in parallel hardware implementations leak similarly. It also suggests that the AES may not be the best block cipher for integration in a leakage-resilient PRF and left a number of questions open regarding the security of this proposal. In this paper, we contribute to these issues in two main directions.

On one hand, we extend the leakage-resilient security analysis of [26], paying attention to three different abstraction levels. At the algorithm level, we investigate generic side-channel attacks targeting the first and second rounds of a re-keying function (and check how much they can help to break the "identical leakage assumption"). We also use our analysis to provide a discussion of the tradeoff between the time and data complexity of such attacks. At the architecture level, we exhibit a possible weakness in the (realistic) case where an implementation would leak according to a distance-based leakage model (e.g. the Hamming distance one). We then put forward different solutions to mitigate the issue. Eventually, at the implementation level, we study the impact of localized Electro-Magnetic Analysis (EMA) [10,30] for distinguishing the leakage of different components of our constructions. We use an FPGA case study to highlight that the resulting (key-dependent) algorithmic noise remains difficult to exploit by actual adversaries.

On the other hand, we take advantage of our security evaluations in order to specify the components of a block cipher that would be better suited to leakage-resilience than the AES. Starting from a PRESENT-like structure [2] (a natural candidate for hardware implementations), the results of our algorithmic-level security analysis allows determining the size of S-boxes in this cipher, while the result of our architecture-level security analysis leads to new criteria for the permutation layer. The latter example is interesting from a methodological point of view, as it suggests that low-level issues in physical security can sometimes be more efficiently solved at higher abstraction levels. We claim that the resulting cipher integrated in the leakage-resilient PRF construction from CHES 2012 can lead to secure and efficient implementations of the fresh re-keying scheme in Figure 1.

Organization. We start the paper with the description of a generic block cipher for use in leakage-resilient schemes, leaving some parameters open (e.g. the previously mentioned S-box size and permutation layer). Following, Sections 3, 4 and 5 contain our security analyzes at different abstraction levels and fix the open parameters progressively. Eventually, we specify an instance of block cipher based re-keying function in Section 6, and detail an open source Hardware Description Language (HDL) code for an instance of hardware architecture. We hope that this open source code will stimulate further research and practical security evaluations of our proposal. Conclusions are in Section 7.

Cautionary note. Our focus is on the side-channel resistance of the proposed construction. In this context, the first/last encryption rounds of a block cipher implementation are usually the most critical. We consequently investigate these rounds as an important step in validating the interest of leakage-resilient PRFs based on block ciphers. By contrast, we *do not* make any specific claim regarding the fact that our proposal is a secure PRF yet. Our hope and belief naturally is that combining enough of the iterations proposed in this paper can lead to mathematical security at lower cost than previous proposals, and our performance evaluations in Section 6 provide good indications that this could indeed be the case. Meanwhile, we specify our constructions up to the point where its physical security can be analyzed, and suggest to use it as a possible instance for the function g in Figure 1, since it has relaxed requirements from the mathematical point of view.

2 Towards efficient leakage-resilient PRF designs

The block diagram of our instance of re-keying function g is given in Figure 2, where $r[0]$ denotes the 0th word of the public random nonce r in Figure 1, and the word size is determined by the S-box size of the underlying cipher used in the re-keying steps. In the CHES 2012 proposal, each step corresponds to the execution of the AES Rijndael and the words are 8-bit wide. In the rest of this paper, we will consider an alternative (generic) cipher design represented in the right part of the picture, in which the iterations combine a bitwise key addition, an S-box layer and a permutation layer (aka wire crossing). Intuitively, the improved physical security of this re-keying function comes from the careful selection of this plaintexts that can be enforced in tree-based PRFs. Namely, the block cipher (i.e. the steps) in Figure 2 can only be queried with inputs of a very specific format, where each word of r bears the same value (i.e. $r[0]||r[0]||\ldots||r[0]$ for the first round). This implies that any divide-and-conquer DPA trying to exploit the leakage will be affected by a key-dependent algorithmic noise. Besides, if the leakage functions corresponding to all the S-boxes are identical, they will only provide information about the master key (e.g. k_0) up to a permutation of its words (see [26] for a detailed analysis of this claim).

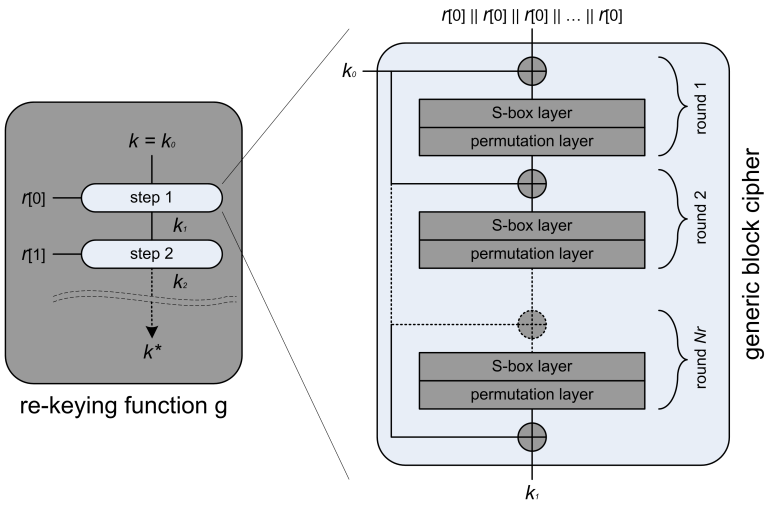

Fig. 2. Our instance of re-keying function g.

As previously mentioned, using this construction for re-keying rather than directly as a PRF (which would require a secure block cipher) allows relaxing its mathematical security requirements, leading to the following advantages. First, the number of rounds in the block cipher can possibly be reduced since this block cipher essentially needs to fulfill the diffusion criteria detailed in [25]. Second, the output of the re-keying function will be used as a fresh session key k^* that is not public. Hence, the output whitening step of the CHES 2012 construction is not necessary. For performance reasons, we will also consider a very minimum key scheduling (inspired by [3,12]), which allows that the recovery of any ith step key k_i does not directly translate into a master key recovery. Given these a priori choices, the main design questions we will consider in the next sections are:

1. How to select the S-boxes number N_s and bit size b?
2. How to choose the permutation layer?
3. How many block cipher rounds per step are necessary?
4. How many steps are necessary?

The analysis of Section 3 will allow answering the first question. The analysis of Section 4 will allow answering the second question. As for the number of rounds and steps, we will discuss minimum requirements for fresh re-keying applications in Section 4.3.

3 Physical security analysis at the algorithm level

In this section, we investigate the physical security of our generic block cipher construction in a simple model where the leakage produced by each S-box is assumed identical. We first refine the security levels provided in [26], by relaxing the simplifying hypothesis that all key words are pairwise different. Our results show that efficient design choices still allow preventing low-complexity attacks targeting the first S-box layer. Next, we focus on the second block cipher round and highlight possible attacks with practical time complexities. Finally, we exhibit in Section 3.3 that despite its limited time complexity, DPA taking advantage of the second-round leakages may remain difficult because of the bounded data complexity that is guaranteed by our leakage-resilient construction.

3.1 Analysis of the first S-box layer

Our substitution layer is composed of N_s b-bit S-boxes operating in parallel, as illustrated in Figure 3 for $N_s = 4$ and $b = 4$. Intuitively, this parallelism combined with a careful selection of the plaintexts improves security against DPA, since an attacker may succeed in recovering the set $\mathcal{S}$ of the N_s key words, but has no information to order them as long as the leakage functions L_i's are identical. As a result, the security analysis of [26] suggests that successful attacks should have at least the (super-exponential) time complexity of an enumeration over N_s S-boxes. Yet, in practice one should additionally consider that several key words in $\mathcal{S}$ may share the same value in $[0...2^b - 1]$. In this case, the optimistic complexity $N_s!$ has to be divided by the (factorial of the) multiplicities of each value in $\mathcal{S}$. Details about the computations of these multiplicities are given in Appendix A. The resulting (improved) attack complexities are given in the left part of Table 1.

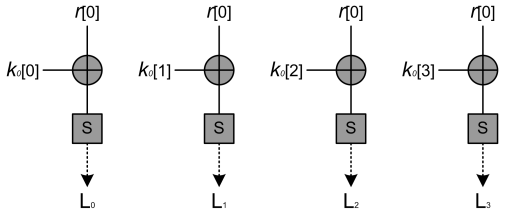

Fig. 3. Attacks against the first S-box layer.

Table 1. Expected time complexities of attacks targeting the first S-box layer (left) and the second S-box layer (right) estimated with Monte Carlo sampling (in $\log_2$ scale).

N_s	16	24	32
$b = 4$	39	66	95
$b = 8$	44	78	116
$b = 12$	44	79	118
$b = 16$	44	79	118

N_s	16	24	32
$b = 4$	13.4	14.8	15.5
$b = 8$	28.8	34.4	38.1
$b = 12$	39.7	50.2	56.5
$b = 16$	44.3	63.7	73.4

3.2 Analysis of the second S-box layer

One important argument in the previous subsection is that it can be computationally difficult to distinguish the different key words in the set $\mathcal{S}$ when the leakage functions L_i's are identical. In this context, a natural question is to know whether the second round leakage could not be used to discriminate these key words with lower complexities. To answer it, we use the exemplary design of Figure 4 (given for $N_s = 4$ and $b = 4$). For now,

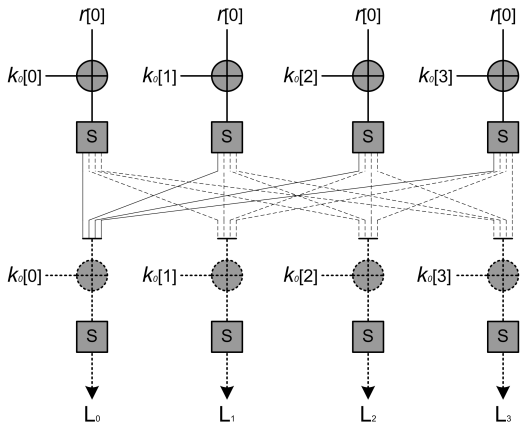

Fig. 4. Attacks against the second S-box layer.

we use the permutation of `Small-Present` in our analysis [18]. In this case, the second-round S-box inputs depend on b key words from the multiset $\mathcal{S}$. So an adversary essentially has to pick these b key words and determine their order. Assuming no key addition in the second round, the first step is equivalent to the enumeration of the b combinations of a multiset of cardinal N_s. Its complexity is given by MacMahon's formula [19]:

$$\sum_{p=0}^{N_s}(-1)^p \sum_{1\leq i_1\leq i_2\leq\cdots\leq i_p\leq N_s} \binom{N_s + b - m_{i_1} - m_{i_2} - \cdots - m_{i_p} - p - 1}{N_s - 1},$$

with the m_i's standing for the multiplicities of the values in $\mathcal{S}$. The complexity of second step is determined as in the previous subsection. The resulting attack complexities are given in the right part of Table 1. Additionally considering a key addition in the second round would multiply them by 2^b. We conclude that the large time complexities obtained when only taking advantage of first round leakages vanish if the second round is targeted.

3.3 Time complexity vs. data complexity tradeoff

Since the best attacks exploiting second round leakages do not have a sufficiently high time complexity for ensuring practical security, we finally investigate the exploitation of this leakage in the context of practical adversaries with data complexity bounded to 2^b, as guaranteed by design in our leakage-resilient construction. In this case, the main goal is to solve the estimation issue that is typical from side-channel distinguishers. We will focus on Brier et al.'s Correlation Power Analysis (CPA) to illustrate our claims [4]. Yet, we note that in a first-order DPA scenario, this distinguisher is actually equivalent to worst-case template attacks as long as both distinguishers use the same leakage models [16]. Since our following analyzes essentially consider perfect leakage models anyway, our conclusions are in fact reflective of most actual strategies that could be used in practice [20].

In general, a successful CPA requires that an adversary can distinguish a correlation coefficient estimated for the correct key candidate (denoted as ρ_g) from correlation coefficients estimated for wrong key candidates (denoted as ρ_w). In order to simplify analyses, a usual assumption is to consider $\rho_w = 0$ (i.e. that wrong key candidates give rise to uncorrelated signals) [22]. Further assuming that the adversary obtains noiseless leakages and that she perfectly knows the leakage model, we can additionally approximate the maximum correlation obtained for the correct key candidate as $\rho_g \approx \frac{1}{\sqrt{N_s}}$. In this simple setting, the number of traces required to distinguish both distributions is given by [21]:

$$N_t = 3 + 8 * \frac{z_{1-\alpha}^2}{\ln^2 \frac{1+\rho_g}{1-\rho_g}}, \tag{1}$$

with $z_{1-\alpha}$ the quantile value. When testing N_k key candidates, we typically set the the confidence α to $\frac{1}{N_k}$. This number of key candidates to test for the attack strategies described in Section 3.2 is given in the left part of Table 2. It directly leads to the minimum data complexities required for a CPA to be successful for various parameters N_s and b, given in the right part of the table. For $b = 4$, $b = 8$ and the combination $b = 12$ $N_s = 32$, the data complexity needed is larger than the available 16, 256 and 4096 tolerated by our construction. For $b = 12$ combined with $N_s \leqslant 24$ and $b = 16$, a sufficient number of traces is available to mount a successful attack. This naturally suggests that $b = 4$ is the preferred solution for security reasons (which comes at the cost of lower performances, since less bits of r will be operated per step in Figure 2). The next sections will stick with this design choice and consider $N_s = 32$ to prevent first-round attacks[3].

[3] Since for $b = 4$, N_t might be not large enough for the formula of Equation 1 to be accurate, we also performed the following experiment. We uniformly sampled a 16-tuple of 4-bit values as hypothesis for the correct key (A) and simulated the observed signal by adding 15 more random 16-tuples to the first one (B). Then, we sampled 2^{16} tuples of 4-bit values for the incorrect key hypotheses (C_i). Finally, we applied a Hamming weight leakage function and calculated the 2^{16} correlation coefficients between (A) and (B), and (B) and (C_i) respectively. The resulting coefficients for the wrong hypotheses lied between -0.85 and 0.85. Furthermore, over 100 experiments we observed that on average 18 000 wrong hypotheses yielded a higher ρ than the correct key. The observed minimum of favored wrong keys was 209 and the maximum 64 800. This experiment identically suggests that a b of no more than four should be chosen.

Table 2. Left: number of key hypotheses to test for a known key words set (in $\log_2$ scale). Right: Minimum number of traces to mount a successful CPA with sufficient confidence.

N_s	16	24	32
$b = 4$	13.4	14.8	15.5
$b = 8$	28.8	34.4	38.1
$b = 12$	39.7	50.2	56.5
$b = 16$	44.3	63.7	73.4

N_s	16	24	32
$b = 4$	432	741	1051
$b = 8$	1060	1966	2954
$b = 12$	1513	2969	4526
$b = 16$	1705	3831	5977

4 Physical security analysis at the architecture level

The previous analyzes are essentially independent of the architecture used to implement our re-keying scheme. In this section, we move towards a lower abstraction level and investigate possible attacks taking advantage of a typical hardware implementation that would implement the operations of our block cipher round in parallel. In this context, an important observation is that CMOS devices usually leak proportional to the Hamming distance of values which appear subsequently in a part of the hardware module, e.g. a data bus or register. As a result, an attack can take advantage of extra information provided by the combined leakage of the two values (which would not be available in two separate attacks on the individual values). We first show that such attacks exist in a reasonable implementation context, and then discuss how to mitigate them with an appropriate choice of permutation layer. Finally, we conclude the section with a short discussion of the minimum number of rounds per step required for secure re-keying.

4.1 The Hamming distance issue

As our leakage-resilient design requires the parallel execution of all the S-boxes, a natural architecture for implementing a re-keying step would consist of a single-round unit performing key addition, substitution and permutation in a single clock cycle, whose result is fed back until the required number of rounds is reached. In a device leaking the Hamming distance, this would mean that there is combined leakage of two values occurring at the same point in subsequent rounds, e.g. two round inputs or two S-box outputs. Assuming that the permutation layer used in the rounds is exactly the one of `Small-Present` as proposed in Section 3.1, such a Hamming distance leakage would directly lead to improved attacks. The main issue is that such a permutation layer has the property that the relative position of a bit within a word after the permutation is dependent on the index of the S-box the bit originated from. For example for $N_s = 4$, the first bit of each word after the permutation originates exclusively from the first S-box. Considering (as in Section 3.1) that the values of the key words are known and only their order remains unknown, an attacker could further identify (e.g.) the first key word in the following way. Derive the S-box output using the value of each key word and calculate the Hamming distance with a word consisting of the first bit of the input replicated four times. When attacking the power traces with these hypotheses, only the one for the actual first key word will succeed. This process can be repeated for all other key word positions using different bits from the nonce word to calculate the Hamming distance hypothesis.

For $N_s > b$, the position of the key word cannot be determined uniquely by this attack. However, their number of possible orderings will be reduced significantly, even in the optimistic case where these words are all pairwise independent. Before the attack, each key word could potentially appear in each position of the key, giving $N_s!$ possible candidates (in this optimistic case). After the attack there will be b mutually exclusive groups of N_s/b candidates, each belonging to fixed parts of the key. So only the ordering within the b groups will remain unknown, leading to $((N_s/b)!)^b$ possible candidates (again in the optimistic case). A straightforward solution to avoid this issue is to deal with it at the architecture level, i.e. design an implementation where such Hamming distance leakages do not appear. For example, one could use multiple registers for this purpose (so that the output of a round never erases its input). In the next subsection, we show that a change of the permutation layer allows mitigating the issue at lower cost.

4.2 Mitigating distance-based leakages with the permutation layer

The described Hamming distance attack is enabled by the structure of the permutation layer. It is therefore interesting to examine alternative permutations which avoid this particular property but retain the desired diffusion properties. This means that for each bit of the output of the new permutation, the offset within its word should not depend on the index of the S-box the bit originated from. Put another way, all S-box output bits of a specific offset (e.g. all first bits of the S-box outputs) should end up in the same position of a word after the permutation (e.g. the first bit of a word). The diffusion of the permutation should still be optimal (as for the permutation of `Small-Present`). Optimal diffusion means that full diffusion (i.e. each output bit depends on each input bit) is reached after at most $\lceil log_b(N_s) \rceil$ rounds of the substitution-permutation network[4].

We have constructed several such permutations. For example, a fairly general variant for arbitrary values of N_s and b (with the requirement $N_s \equiv 0 \pmod b$) is given by:

$$P(i) = ((i \bmod b) * (N_s + 1) + (\lfloor i/b \rfloor \bmod b) * N_s + \lfloor i/b^2 \rfloor * b) \bmod (b * N_s).$$

This permutation connects the first bit of each S-box output to the first bit of a word after the permutation, the second bit of each S-box output to the second bit of a word after the permutation, ... Hence, an attack using the Hamming distance as described in Section 4.1 yields no extra information about the location of the key words.

4.3 Number of rounds per step

In order to keep our construction efficient, it is naturally desirable to minimize the number of rounds per step. In this respect, let us assume that an adversary can use two consecutive chunks of r to recover the input and the output of a step up to a permutation over the S-boxes. If one step does not have full diffusion (e.g. if it has too few rounds), she should again be able to exclude some positions for the key bytes and thus reduce the complexity of finding their order. By contrast, a step with full diffusion would then require to guess the permutation in the first place (so that nothing can be gained by such an attack anymore). In the following, we will consequently set as minimum criterion that one step should have

[4] Under the assumption that the S-box does not contain structural weaknesses.

complete diffusion. By using a permutation with optimal diffusion, $\lceil log_b(N_s) \rceil$ rounds are necessary to reach full diffusion. For 4-bit S-boxes ($b = 4$), a choice of $4 < N_s \leq 16$ would require at least two rounds and $16 < N_s \leq 64$ would require at least three rounds per step. A security margin of one or two rounds could be added depending on the number of S-boxes. Such parameters are sufficient for ensuring the security of the re-keying scheme in Figure 1 (since they fulfill the six conditions stated in [25]). Besides, we note that they also provide a better mathematical security level than the modular multiplication of the Africacrypt 2010 proposal (e.g. some non-linearity is provided by the use of S-boxes). As mentioned in introduction, it is an interesting open problem to determine the number of rounds required for our construction to become a mathematically strong PRF.

5 Physical security analysis at the implementation level

In this section, we further move down to low abstraction levels and investigate the practicality of the "similar leakage" assumption that is probably the most important one to validate in practice. For this purpose, we consider a prototype implementation of our leakage-resilient construction on a FPGA, and evaluate its security against localized Electro-Magnetic (EM) field analysis [13], which was left as an open problem in [26]. The architecture of the design is detailed in Appendix B. It implements the first round of our construction in the first step (as described in Section 2), using 32 parallel PRESENT S-boxes and the permutation layer presented in Section 4.2. In order to allow worst-case analysis of our re-keying function, the architecture additionally provides two operational modes. In a first (open) mode, it is possible to change each single word of both the master key k and the random nonce r, keeping all the other words constant. While this is exactly what is prevented by construction (i.e. only carefully selected plaintexts are observable by the adversary), this mode was investigated in order to allow profiling without the impediment of the key-dependent algorithmic noise. In the second (fixed) mode, the master key is fixed and the word $r[i]$ of the nonce in the i^{th} step is replicated 32 times to cover the length of the nonce register. This corresponds to the actual circumstances that an adversary would face when attacking our leakage-resilient construction. In the rest of the section, we describe the worst-case profiling together with the selection of points of interests in the EM maps. Next, we present the results of attacks against our implementation in fixed mode taking advantage of these worst-case profiles. Finally, we discuss the practical relevance of worst-case evaluations and the interpretation of our results.

5.1 Worst-case profiling in open mode

In open mode, the adversary is able to independently observe the EM leakage characteristic of each subkey at different locations over the chip surface, without the influence of the key-dependent algorithmic noise (since the untargeted words can be set to random values). Hence, she can directly profile a leakage model of each subkey, just as in any other parallel implementation. In order to identify the univariate leakage of individual subkeys, we recorded 2^{16} measurements and computed the signal-to-noise ratio (SNR) for each word j, at each location (x, y) and for each time instant t. That is, $\mathrm{SNR}_j(x, y, t) = \frac{\hat{\sigma}(\hat{\mu}_{0\to 0}, \hat{\mu}_{0\to 1}, \ldots, \hat{\mu}_{F\to F})}{\hat{\mu}(\hat{\sigma}_{0\to 0}, \hat{\sigma}_{0\to 1}, \ldots, \hat{\sigma}_{F\to F})}$, where $\hat{\mu}_{u\to w}$ and $\hat{\sigma}_{u\to w}$ are the maximum likelihood estimators of the mean value and standard deviation of the leakages at time instant

t conditioned on the transition from the value u to the value w of the target S-box. The 4-bit inputs to the key and nonce registers were carefully chosen from a 16-bit LFSR, in order to produce all the possible 256 transitions of a word in the state register exactly 256 times each. As a result, we obtained 32 SNR maps which are provided in Appendix C. It can be observed that the leakage of individual key words are clearly bounded to some confined regions on the chip surface. However, if we consider the leakage of each subkey as occurring simultaneously during an actual attack, then all the SNRs overlap significantly, as shown in the left part of Figure 5. This result can be easily explained by looking at the placement of our design on the floorplan, which is shown in the right part of the figure. In fact, contrary to [14] where constraints on the placement were set, in our case the logic cells on the floorplan of the FPGA are located only in one large fuzzy region (due to an unconstrained placement). This region overlaps with the region of high SNR.

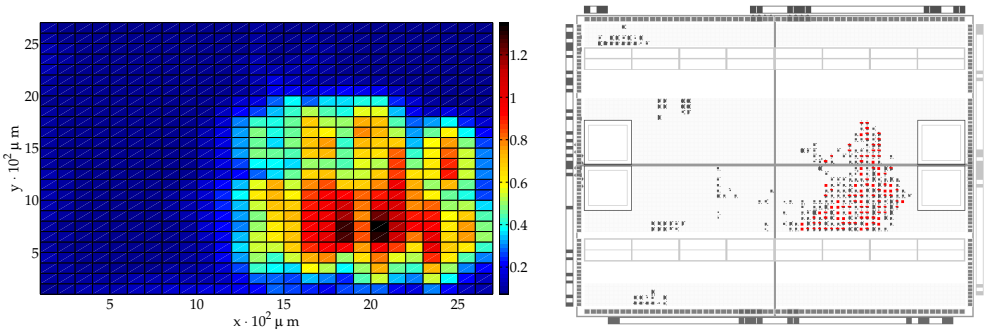

Fig. 5. Left: SNR over the 27×27 chip surface. Right: Placement on the floorplan

Given these preliminary results, the next question is to determine how to select the Points Of Interests (POIs) that will be used in our attacks. Quite naturally, the previous SNRs considered individually are not optimal in this respect, since they are based on the implicit assumption of independent (algorithmic) noise. Therefore, we considered two additional criteria in order to better reflect the activity of individual key words considering the presence of key-dependent algorithmic noise, namely:

$$C_2 = \max \frac{\mathrm{SNR}_j(x,y,t)}{\sum_{i \neq j} \mathrm{SNR}_i(x,y,t)}, \quad C_3 = \max \frac{\mathrm{SNR}_j(x,y,t)}{\max_{i \neq j} \mathrm{SNR}_i(x,y,t)}. \tag{2}$$

The intuition behind these criteria is that the best POIs should isolate one target S-box from either all the other S-boxes (on average) or from the "closest" S-box.

5.2 Attacks exploiting worst-case profiles in fixed mode

For the different selections of POIs in the previous section (including the basic SNR), we built leakage models and then performed 32 CPA attacks in fixed mode (i.e. for a fixed key, with the nonces defined in Section 2), using a fresh set of measurements. In this context, the data complexity for each attack is bounded to 16. Yet, nothing prevents an adversary to repeatedly measure each of its allowed input queries in order to get rid of

physical noise. Hence, we performed attacks exploiting increasing number of traces (from 2^8 to 2^{16}) and first observed that the results were stable from 2^{12} traces on. Next, we had a look at the subkey ranks (i.e. the position of the correct subkeys in the 32 vectors of 16 candidates as provided by the attacks). For illustration, we list the ones obtained for the worst criteria (SNR) and the best one (i.e. C_2 or C_3 depending on the S-boxes):

Subkey ranks (SNR): [1 5 14 7 6 8 3 1 2 1 1 14 14 1 7 1 6 9 6 15 6 1 1 3 6 16 7 14 8 2 11 1].
Subkey ranks (best): [1 1 14 2 3 4 1 1 2 1 1 7 14 1 7 1 6 6 6 12 1 1 1 3 3 6 2 12 8 1 7 1].

One can directly observe that for a number of S-boxes (namely 9 for the worst and 13 for the best cases), the correct subkey is ranked first - hence suggesting that the localized EM profiling indeed allows improved attacks. Yet, looking at the CPA results more precisely, we also observed that firstly ranked subkeys were usually slightly better correlated than other candidates. By contrast for badly ranked subkeys, some of them showed very poor correlation results. The main consequence of this observation is that enumerating the master key remains a computationally intensive task, even in the context where worst-case profiling is possible. To illustrate this claim, first observe that an underestimated time complexity for the enumeration can be obtained by computing the product of the subkey ranks. From the two previous lists, we obtain 2^{64} and 2^{46}, respectively. Improving this lower bound can be done by merging the lists, e.g. the 16 subkey ranks for 8-bit bytes corresponding to the same two attacks (aggregated) are given by:

Subkey ranks (SNR): [9 202 59 9 7 68 78 26 90 159 6 11 142 112 80 78],
Subkey ranks (best): [1 76 19 1 7 27 78 26 50 107 1 11 35 50 43 36],

leading to refined bounds of 2^{86} and 2^{66}, respectively. Intuitively, the better bounds derive from the fact that when merging dimensions (as an optimal key enumeration algorithm does [37]), the time complexity significantly increases every time both subkeys are not highly ranked. Using the rank estimation algorithm in [38], we finally obtained tight bounds for the master key rank as $[2^{115} - 2^{118}]$ and $[2^{99} - 2^{102}]$. Quite naturally, one could further consider that the knowledge of which subkeys are "easy to recover" is an additional outcome of the worst-case profiling[5]. In this conservative scenario, the adversary could reduce the dimension of her enumeration problem (down to 23 and 19, respectively), but our experiments still lead to security bounds of $[2^{89} - 2^{90}]$ and $[2^{69} - 2^{70}]$.

5.3 Interpretation of the results

The previous results are encouraging, as they suggest that the master key of our construction remains hard to enumerate, even in conditions where worst-case profiling is possible. Despite being difficult to compare (since based on totally different hardware assumptions), it is worth to note that under similar conditions, the security of a masked implementation would most likely be quite weak (since the localized electromagnetic measurements would allow obtaining low-noise leakages for each of the shares [36]). Nevertheless, it is also important to consider these results with care, as they only correspond to a single implementation context. In this respect, we emphasize the large number of factors that could have impact on our conclusions, such as the manufacturing technology, the distribution of active logic cells on the floorplan, the resolution of the coil, and the distance

[5] This is realistic as this information mainly depends on the placement of the S-boxes in the implementation. By contrast, the information of the correct subkey ranking depends on the key-dependent algorithmic noise and cannot be considered as constant for all attacks.

and materials between the probe and the leaking circuitry. We now briefly discuss the interpretation of our experiments with respect to two important axes, namely (i) what are the possible improvements and (ii) how representative is worst-case profiling.

As far as improvements are concerned, they could certainly go in two directions. On the one hand, improved attacks could be considered. The most natural proposal would be to take advantage of multivariate leakages in order to better discriminate the S-boxes. It raises many interesting open problems. For example, the selection of POIs could not be based on SNRs anymore in this case. Best exploiting multivariate information would require to perform the information theoretic evaluations advertised in [34] and to exploit dimensionality reductions such as, e.g. [1,33]. These evaluations may turn out to become challenging in view of the huge data sets considered in our experiments (more than one week of measurements and 400GB of traces). On the other hand, several solutions to improve the countermeasure could be studied as well. In this respect, a starting observation is that the discrimination of S-boxes in our leakage-resilient construction inherently requires some profiling. Therefore, general questions about the portability of templates (e.g. in front of nanoscale devices with variability [31]) are particularly relevant in this case. Besides, the investigation of place-and-route constraints that best allow "interleaving" the S-boxes in our design is certainly another interesting scope for further research. Moving from FPGAs to ASICs could also reveal additional opportunities to improve the countermeasure. Eventually and if needed, taking advantage of space randomization such as proposed, e.g. in [28], is yet another possible track for security enhancement.

As far as worst-case profiling is concerned, the main question is whether similar results could be obtained in the more realistic scenario where the implementation is in fixed mode for profiling as well. One direct problem is that in this context, the first-round leakages cannot be exploited as in this section. In fact, the transitions used to compute our SNRs would all be equivalent up to a permutation in this case, making it impossible to select POIs for different subkeys. Nevertheless, alternative profiling paths also exist. One solution would be to "group" similar transitions thanks to a non-bijective transformation. But the choice of a transformation that adequately captures the similarity of different transitions is not straightforward (and we can anyway only loose information by profiling in this way). For example, experiments performed under a Hamming distance transformation exhibited significantly reduced SNRs for our prototype. Another solution is to profile second-round leakages. But this would require building more templates and could also be limited by the bounded data complexity issues discussed in Section 3.2. Other options certainly exist and are an interesting scope for further research. Meanwhile, we conclude that although fixed-key profiling may be more annoying to perform in practice, considering worst-case (open) profiling for reference is certainly a relevant choice for the evaluation of our countermeasure in view of the improved attacks that could be designed.

6 An open source and generic VHDL code

In order to estimate the costs of our method in terms of speed and size in silicon, we implemented a leakage-resilient re-keying function in VHDL. We decided to keep the design as flexible as possible to allow realizing and testing different configurations. The parameters a designer can set before synthesizing the re-keying function comprise the

number of rounds within a step and the number of steps to generate a fresh key. Furthermore, both 4-bit PRESENT S-boxes and 8-bit AES S-boxes can be selected. The designer can additionally choose the desired bit-size of the data path and hence the size of the key-material generated. Finally, and as a complement to functional parameters, a tradeoff between speed and required area can be configured. That is, the implementation supports unrolling of rounds, where the overall number of rounds must be a multiple of those performed in a single clock cycle. Thus, the latency to generate a fresh key using our construction can be computed as (number of steps) $*$ (number of rounds) / (unrolled rounds) + (one initialization cycle). An overview of this architecture is given in Figure 6. The dotted lines in the figure depict possible extensions of the design that were not used in this paper. For example, the design is ready for including a final step like a Davies-Meyer transformation and/or a key expansion that transforms the key between each round.

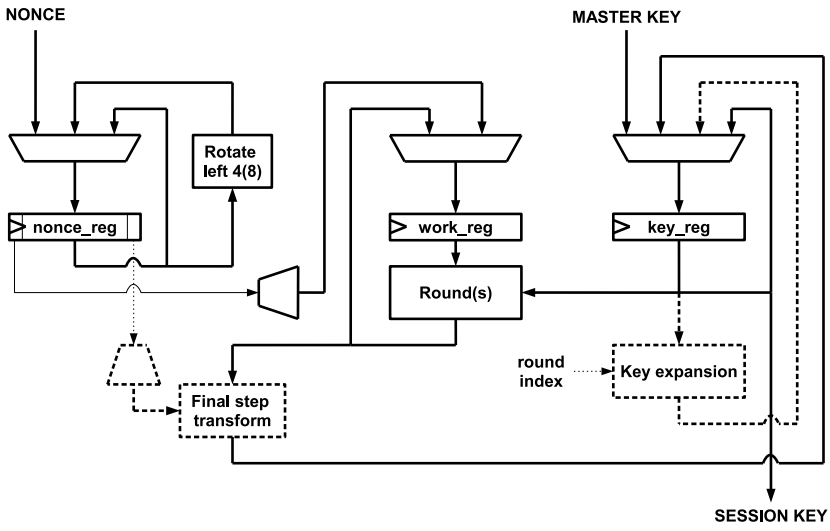

Fig. 6. Overview of our open source hardware architecture for fresh re-keying.

Our synthesis results for implementations with different configuration options are shown in Table 3. We targeted the UMC 0.18 μm FSA0A_C standard-cell library [8] and did not perform timing optimizations. The first two implementations use our recommended configuration with two different degrees of round unrolling, while the third implementation features the absolute minimal options which could still yield a moderate degree of security. All implementations use the PERSENT S-box, the linear layer proposed in Section 4.2, no key expansion, and no final transformation in the step[6].

It is important to note that 562 cycles for 7,300 gate equivalents correspond to the cost of a first-order masked implementation for the modular multiplication in [25]. So the performances of our architecture already compare favorably with this one (moving to higher-order masking naturally makes the comparison even better). Besides and most

[6] Our source codes are available under an open source license on the authors' home pages.

importantly, our implementation is a parallel one while the Africacrypt 2010 one is only 8-bit wide. This means that reaching acceptable noise levels for the masking countermeasure to become effective requires additional shuffling, e.g. as proposed in [24] and leading to significant performance overheads (in the 10th of thousands cycles). These preliminary investigations suggest with good confidence that in a hardware context, the fresh re-keying based the construction we describe in this paper had good potential to lead to a better performance vs. security tradeoff than a masked modular multiplication.

Table 3. Synthesis results using the UMC 0.18 μm FSA0A_C standard-cell library.

S-boxes/steps/rounds/unrolled rounds	**Latency** cycles	**Area** GE	**Clock freq.** MHz
32/32/5/1	161	7,300	338
32/32/5/5	33	16,828	210
24/20/3/1	61	5,302	354

7 Conclusion

In this paper, we presented an exploratory analysis of the design space for fresh re-keying opened by the use of leakage-resilient PRFs to prevent side-channel attacks. Our results provide new guidelines for the choice of block cipher components to consider in this context and for their implementation. Admittedly, the understanding and security evaluation of this type of constructions is still far from the one of standard protections such as masking and shuffling. Yet, the preliminary investigations we describe in this paper are promising and suggest new solutions to build physically secure hardware devices. From the side-channel security point of view, further optimizing the localized electromagnetic measurements bu exploiting multivariate attacks is certainly worth further investigations. Depending on the strength of these advanced attacks, space-randomized implementations (where the localization of the S-box executions would vary over time) could then be designed as well. From a more theoretical point of view, it would be interesting to investigate whether a PRF could be directly obtained by extending the number of rounds of our new construction. It would allow to use it directly as a leakage-resilient primitive rather than for re-keying the AES, and maybe to obtain additional performance gains.

Acknowledgements. This work has been funded in part by the European Commissions ECRYPT-II NoE (ICT-2007-216676) and by the 7th framework European project TAMPRES, by the ERC project 280141 (acronym CRASH). François-Xavier Standaert is a Research Associate of the Belgian Fund for Scientific Research (FNRS-F.R.S).

References

1. Cédric Archambeau, Eric Peeters, François-Xavier Standaert, and Jean-Jacques Quisquater. Template attacks in principal subspaces. In Louis Goubin and Mitsuru Matsui, editors, *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2006.
2. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.
3. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, François-Xavier Standaert, John P. Steinberger, and Elmar Tischhauser. Key-alternating ciphers in a provable setting: Encryption using a small number of public permutations - (extended abstract). In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 45–62. Springer, 2012.
4. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.
5. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.
6. Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In Tal Rabin, editor, *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.
7. Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008.
8. Faraday Technology Corporation. Faraday FSA0A_C 0.18 μm ASIC Standard Cell Library, 2004. Details available online at `http://www.faraday-tech.com`.
9. Sebastian Faust, Krzysztof Pietrzak, and Joachim Schipper. Practical leakage-resilient symmetric cryptography. In Prouff and Schaumont [29], pages 213–232.
10. Karine Gandolfi, Christophe Mourtel, and Francis Olivier. Electromagnetic analysis: Concrete results. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *CHES*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.
11. Louis Goubin and Jacques Patarin. Des and differential power analysis (the "duplication" method). In Çetin Kaya Koç and Christof Paar, editors, *CHES*, volume 1717 of *Lecture Notes in Computer Science*, pages 158–172. Springer, 1999.
12. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The led block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011.
13. Johann Heyszl, Stefan Mangard, Benedikt Heinz, Frederic Stumpf, and Georg Sigl. Localized electromagnetic analysis of cryptographic implementations. In Orr Dunkelman, editor, *CT-RSA*, volume 7178 of *Lecture Notes in Computer Science*, pages 231–244. Springer, 2012.
14. Johann Heyszl, Dominik Merli, Benedikt Heinz, Fabrizio De Santis, and Georg Sigl. Strengths and limitations of high-resolution electromagnetic field measurements for side-channel analysis. In Stefan Mangard, editor, *Smart Card Research and Advanced Applications*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012.
15. Antoine Joux, editor. *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*. Springer, 2009.
16. Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors. *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2003.

17. Paul C. Kocher. Leak resistant cryptographic indexed key update. US Patent.
18. Gregor Leander. Small scale variants of the block cipher present. Cryptology ePrint Archive, Report 2010/143, 2010. http://eprint.iacr.org/.
19. Percy Alexander MacMahon. *Percy Alexander MacMahon: Collected Papers - Vol. 1: Combinatorics*. MIT Press, 1978.
20. S. Mangard, E. Oswald, and F.-X. Standaert. One for all – all for one: unifying standard differential power analysis attacks. *IET Information Security*, 5(2):100–110, 2011.
21. Stefan Mangard. Hardware countermeasures against dpa ? a statistical analysis of their effectiveness. In Tatsuaki Okamoto, editor, *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.
22. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.
23. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In Alfred Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.
24. Marcel Medwed, Christophe Petit, Francesco Regazzoni, Mathieu Renauld, and François-Xavier Standaert. Fresh re-keying ii: Securing multiple parties against side-channel and fault attacks. In Emmanuel Prouff, editor, *CARDIS*, volume 7079 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2011.
25. Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In Daniel J. Bernstein and Tanja Lange, editors, *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.
26. Marcel Medwed, François-Xavier Standaert, and Antoine Joux. Towards super-exponential side-channel security with efficient leakage-resilient prfs. In Prouff and Schaumont [29], pages 193–212.
27. Krzysztof Pietrzak. A leakage-resilient mode of operation. In Joux [15], pages 462–482.
28. François Poucheret, Lyonel Barthe, Pascal Benoit, Lionel Torres, Philippe Maurine, and Michel Robert. Spatial EM jamming: A countermeasure against EM analysis? In *VLSI-SoC*, pages 105–110. IEEE, 2010.
29. Emmanuel Prouff and Patrick Schaumont, editors. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*. Springer, 2012.
30. Jean-Jacques Quisquater and David Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In Isabelle Attali and Thomas P. Jensen, editors, *E-smart*, volume 2140 of *Lecture Notes in Computer Science*, pages 200–210. Springer, 2001.
31. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A formal study of power variability issues and side-channel attacks for nanoscale devices. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2011.
32. Matthieu Rivain and Emmanuel Prouff. Provably secure higher-order masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 413–427. Springer, 2010.
33. François-Xavier Standaert and Cédric Archambeau. Using subspace-based template attacks to compare and combine power and electromagnetic information leakages. In Elisabeth Oswald and Pankaj Rohatgi, editors, *CHES*, volume 5154 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2008.
34. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In Joux [15], pages 443–461.
35. François-Xavier Standaert, Olivier Pereira, Yu Yu, Jean-Jacques Quisquater, Moti Yung, and Elisabeth Oswald. Leakage resilient cryptography in practice. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 99–134. Springer Berlin Heidelberg, 2010.

36. François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order dpa. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 112–129. Springer, 2010.
37. Nicolas Veyrat-Charvillon, Benoit Gerard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. Cryptology ePrint Archive, Report 2011/610, 2011. http://eprint.iacr.org/.
38. Nicolas Veyrat-Charvillon, Benoit Gerard, and Francois-Xavier Standaert. Security evaluations beyond computing power. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 126–141. Springer Berlin Heidelberg, 2013.
39. Yu Yu and François-Xavier Standaert. Practical leakage-resilient pseudorandom objects with minimum public randomness. In Ed Dawson, editor, *CT-RSA*, volume 7779 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2013.
40. Yu Yu, François-Xavier Standaert, Olivier Pereira, and Moti Yung. Practical leakage-resilient pseudorandom generators. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 141–151. ACM, 2010.

A Impact of key words repetitions

Let us denote by $\mathcal{S}$ a multiset of N_s key words uniformly distributed in $[0..2^b - 1]$. The number of permutations of these key words, or equivalently the complexity to order them, depends on the multiplicities of these key words in $\mathcal{S}$. We denote by m_j (with $1 \leq j \leq 2^b - 1$) the multiplicity of value j, i.e. the number of times this value appears in the multiset $\mathcal{S}$. For instance, with $\mathcal{S} = \{3, 3, 5, 8, 8, 8\}$ ($N_s = 6$), we have $m_3 = 2$, $m_5 = 1$, $m_8 = 3$ and $m_j = 0$, $\forall j \in [0, 2^4 - 1] \backslash \{3, 5, 8\}$. Let us additionally denote by M_i^q the random variable representing the number of multiplicities equal to q when selecting the i^{th} key word (with $1 \leqslant i \leqslant N$). We can then write the following recursion formula that, under relevant boundary conditions, gives us the desired probabilities:

$$\begin{aligned}\forall i, q, k \in [0..N_s],\ \mathrm{P}\left[M_{i+1}^q = k\right] = {} & \frac{k+1}{2^b}\ \mathrm{P}\left[M_i^q = k+1\right] \\ & + \sum_{l=0}^{N} \Big(\mathrm{P}\left[M_i^q = k-1\right]\ \frac{l}{2^b} \\ & + \mathrm{P}\left[M_i^q = k\right]\ (1 - \frac{k+l}{2^b}) \Big)\ \mathrm{P}\left[M_i^{q-1} = l\right].\end{aligned}$$

From these probabilities, we can deduce those of the time complexities of attacks for various parameters N_s and b. In practice, we used Monte Carlo sampling to evaluate the mean complexities thanks to the multiplicities distribution. That is, we drew a large (i.e. sufficient to have accurate estimates) number of independent random variables following a specific law to estimate its expectation using the law of large numbers.

B Architecture's Design on a FPGA

Our analysis was conducted on a Xilinx Spartan 3 FPGA device manufactured in a 90 nm technology. We performed localized magnetic field measurements using a coil with a resolution of 100 μm very closely positioned to the depackaged circuit's front side surface. We performed 27×27 measurements covering the surface area confined by the conjunctions of the bonding wires. The architecture of the design is shown in Figure 7.

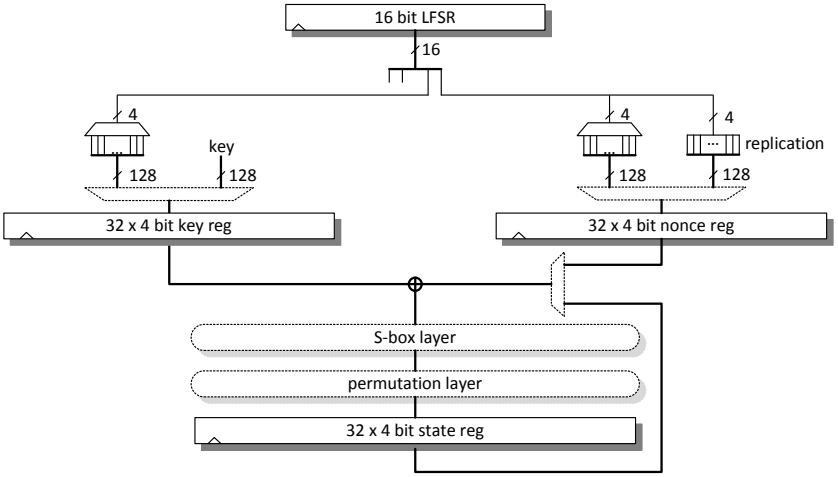

Fig. 7. Prototype architecture for worst-case EM profiling.

C SNR maps of the 32 key words over the chip surface