Exploring Crypto-Physical Dark Matter and Learning with Physical Rounding

Towards Secure and Efficient Fresh Re-Keying

Sébastien Duval, Pierrick Méaux,
Charles Momin, and François-Xavier Standaert

UCLouvain, ICTEAM, Crypto Group, Louvain-la-Neuve, Belgium
firstname.lastname@uclouvain.be

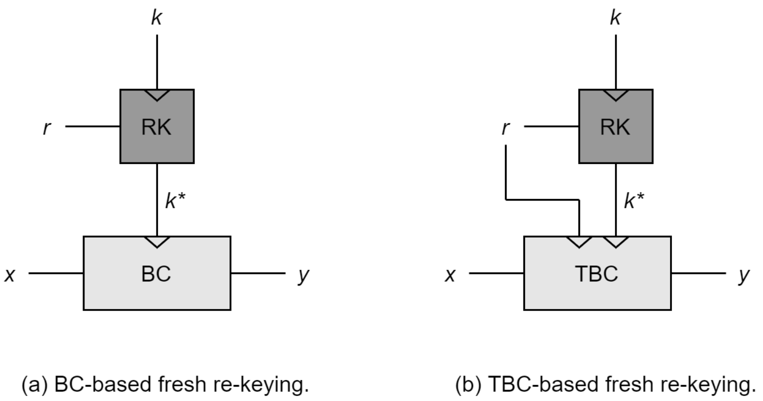
Abstract. State-of-the-art re-keying schemes can be viewed as a tradeoff between efficient but heuristic solutions based on binary field multiplications, that are only secure if implemented with a sufficient amount of noise, and formal but more expensive solutions based on weak pseudorandom functions, that remain secure if the adversary accesses their output in full. Recent results on "crypto dark matter" (TCC 2018) suggest that low-complexity pseudorandom functions can be obtained by mixing linear functions over different small moduli. In this paper, we conjecture that by mixing some matrix multiplications in a prime field with a physical mapping similar to the leakage functions exploited in side-channel analysis, we can build efficient re-keying schemes based on "crypto-physical dark matter", that remain secure against an adversary who can access noise-free measurements. We provide first analyzes of the security and implementation properties that such schemes provide. Precisely, we first show that they are more secure than the initial (heuristic) proposal by Medwed et al. (AFRICACRYPT 2010). For example, they can resist attacks put forward by Belaid et al. (ASIACRYPT 2014), satisfy some relevant cryptographic properties and can be connected to a "Learning with Physical Rounding" problem that shares some similarities with standard learning problems. We next show that they are significantly more efficient than the weak pseudorandom function proposed by Dziembowski et al. (CRYPTO 2016), by exhibiting hardware implementation results.

Keywords: Side-Channel Attacks · Fresh Re-Keying · Low-Complexity wPRFs · Learning With Rounding · Boolean Functions · Masking · Key-Homomorphism

1 Introduction

State-of-the-art. Protecting block cipher implementations against side-channel attacks is a difficult problem. Countermeasures like masking [CJRR99, ISW03] are expensive in software [GR17] and hardware [GMK17]. They are also error prone due to physical defaults such as glitches [MPG05, NRS08] or transitions [CGP$^+$12, BGG$^+$14], and due to composability issues [CPRR13, BBD$^+$16]. Informally, this situation is caused by the complex (nonlinear) nature of the block ciphers: while the linear parts of an implementation can be trivially secret-shared with limited complexity overheads, the secure execution of their nonlinear parts typically implies overheads that are quadratic in the number of shares and requires refreshing algorithms that increase their randomness cost.

As a result of these limitations, the concept of fresh-rekeying (illustrated in Figure 1) was introduced by Medwed et al. [MSGR10]. Its main underlying idea is to leverage a separation of duties between a "re-keying function" RK, that is easy to protect against side-channel attacks (e.g., easy to mask) and is only used to produce a fresh key k^*, and a cryptographically strong function (e.g., a block cipher or a tweakable block cipher) to



(a) BC-based fresh re-keying.

(b) TBC-based fresh re-keying.

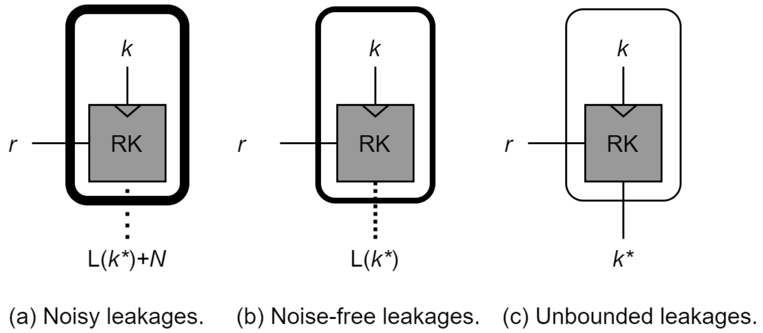
Figure 1: Fresh re-keying schemes: the dark grey blocks must be strongly protected against side-channel attacks, the light grey blocks only require weak protections.

process the messages with this fresh key. As discussed in [DEMM14, DKM$^+$15], the block cipher based solution of Figure 1(a) only provides birthday security and the tweakable block cipher based solution of Figure 1(b) provides beyond-birthday security.

Theoretically, block ciphers and tweakable block ciphers are well understood primitives, with known instances that are secure and efficient (as long as they do not require strong protections against side-channel attacks). By contrast, specifying the properties of the re-keying function turned out to be challenging, and requires adjusting the tradeoff between the re-keying function's efficiency and the physical assumptions needed for its secure implementation, which is reflected by currently published solutions.

Starting with the efficient side of the spectrum, the first proposal of Medwed et al. was a heuristic one, based on a list of necessary properties for the re-keying function. These properties include good diffusion and simplicity to protect against side-channel attacks thanks to masking. Concretely, the instance chosen in [MSGR10, MPR$^+$11] is a finite field multiplication ($k^* = k \cdot r$ over $\mathbb{F}_{2^\kappa}$), which is easy to mask since key-homomorphic. It was next shown by Belaid et al. that in the standard case where Hamming weight leakages are observed by an adversary, a lack of physical noise makes this solution easy to break. Indeed, the least significant bit of the Hamming weight of a value is the XOR of its bits. Hence, for a known r, the least significant bit of $\mathsf{HW}(k \cdot r)$ is a linear function of the bits of k. After κ leakages on average, solving the resulting system of equations therefore leads to the full master key [BFG14]. This attack was then improved in order to remain effective with lower Signal-to-Noise Ratios (SNR) in [BCF$^+$15, PM16, GJ19] and it was shown in the first part of [DFH$^+$16] that the "Learning Parity with (Gaussian) Leakage" (LPL) problem on which this fresh re-keying scheme relies indeed requires significant noise levels (i.e., more than what is available intrinsically in unprotected implementations).

At the other side of the spectrum, Dziembowski et al. studied the possibility to use a weak Pseudo-Random Function (wPRF) as re-keying function [DFH$^+$16]. Their proposal is to perform (several) inner product computations $\langle \boldsymbol{k}, \boldsymbol{x} \rangle$ where $\boldsymbol{k}$ and $\boldsymbol{x} \in \mathbb{Z}_{2^q}^n$ and to "round" the results by computing $\lfloor \langle \boldsymbol{k}, \boldsymbol{x} \rangle \rfloor_\rho$ which simply drops $q - \rho$ bits. The security of this re-keying can be reduced to a Learning With Rounding (LWR) assumption [BPR12, AKPW13], and it is almost key-homomorphic: only the carry bits involved in the shared computations break the key-homomorphism. Hence, by complementing this re-keying with a light error correction scheme (the cost of which increasing logarithmically with the number of shares), this wPRF can be efficiently masked. Yet, despite complexity overheads that are close to linear in the number of shares, such a solution suffers from the large key



(a) Noisy leakages. (b) Noise-free leakages. (c) Unbounded leakages.

Figure 2: Fresh re-keying adversarial models (from weaker to stronger).

size that it requires (e.g., 128-bit security is conjectured for a key of $n = 128 \times q = 32$ bits), the manipulation of which leading to relatively poor performances.

Contribution. The starting point of this work is the observation that state-of-the-art re-keying schemes work in quite different adversarial models. Namely, on the top of the re-keying function internals that have to be protected against side-channel attacks:

- The heuristic solution of Medwed et al. assumes that the adversary can only observe the noisy leakages of the fresh key, as illustrated in Figure 2(a).
- The wPRF-based solution rather assumes that the adversary can observe the fresh key in full (i.e., unbounded leakages), as illustrated in Figure 2(c).

By contrast, no current proposal takes advantage of the intermediate model of Figure 2(b) where noise-free leakages are observed. We argue that recent results on low-complexity PRFs, in particular the one of Boneh et al. on "exploring crypto dark matter" [BIP+18], suggest secure and efficient re-keying schemes could be obtained in this model.

Cryptographic dark matter. The wPRF candidate of Boneh et al. is instantiated as follows. First, it takes a secret matrix $\boldsymbol{K} \in \mathbb{F}_2^{m\times n}$ (possibly Toeplitz for efficiency) and a public vector $\boldsymbol{r} \in \mathbb{F}_2^n$. Next, it computes the product $\boldsymbol{K} \cdot \boldsymbol{r}$ and it interprets the output of this product as a vector of 0/1 values over $\mathbb{F}_3$. Finally, the output of the wPRF is the sum of these values modulo 3. In other words, their wPRF can be defined as:

$$\mathsf{F}_{\boldsymbol{K}}(\boldsymbol{r}) := \mathsf{map}(\boldsymbol{K} \cdot \boldsymbol{r}), \quad (1)$$

with $\mathsf{map} : \{0,1\}^m \to \mathbb{F}_3$ that maps $\boldsymbol{y} \in \{0,1\}^m$ to $\sum \boldsymbol{y}_i \bmod 3$. The similarity between this function and the one of Dziembowski et al. is striking: the matrix multiplication can be seen as multiple inner products and the mapping function plays the role of the rounding. Its limitations for masked implementations are therefore similar: the mapping function is nonlinear over $\mathbb{F}_2^n$ which requires special care and implies overheads.

Crypto-physical dark matter and learning with physical rounding. The main research question we tackle in this paper is whether we can build a secure re-keying scheme in the adversarial model of Figure 2(b), by leveraging some "crypto-physical dark matter". By this, we mean building a wPRF combining a matrix multiplication with a physical mapping that would not have to be computed explicitly (i.e., digitally) and would rather be performed in an analog manner by an implementation's leakages. Taking the example of the Hamming weight function, which is a frequently observed leakage model [MOP07] and will be our running example, we know from the results of Belaid et al. that multiplications in $\mathbb{F}_{2^\kappa}$ make this proposal insecure (without noise). In the following, we put forward that multiplications in a prime field can lead to secure (and efficient) candidates.

Our contributions in this respect are threefold. First, we show that crypto-physical dark matter cannot be secure if its underlying multiplications take place in a small field or if it is based on too small vectors. Next, we propose instances based on medium size prime fields $\mathbb{F}_p$ (e.g., with $p \approx 2^{32}$) that are well suited for software and hardware implementations, and we analyze some relevant security properties of the functions combining prime field multiplications and the Hamming weight mapping. Finally, we highlight the excellent implementation properties that a re-keying scheme based on such a crypto-physical dark matter enables. Informally, these properties are due to the fact that contrary to re-keying schemes in the model of Figure 2(c) where the mapping/rounding has to be computed securely (e.g., thanks to masking), the (physical) mapping/rounding we introduce never has to be computed securely in the model of Figure 2(b), since it is performed by a leakage function. As a result, masking with a small key and complexity overheads that are linear in the number of shares can theoretically be obtained with practically-relevant leakage functions, also leading to a set of interesting open problems browsed in conclusions.

We note that, as usual when introducing a new cryptographic primitive, our focus in this work is to exhibit relevant security & implementation properties which may open new research directions. In this respect, our claim is that the proposed re-keying scheme is at the same time more secure than the one of Medwed et al. [MSGR10] under reasonable (e.g., Hamming weight) leakage models and more efficient than the one of Dziembowski et al. [DFH$^+$16] thanks to a significantly shorter key. We hope these results can be used as a seed to trigger more cryptanalytic investigations and physical security analyzes.

We additionally note that we will use the term crypto-physical dark matter for the re-keying operations and the term Learning With Physical Rounding (LWPR) for the problem of recovering the long-term key of the resulting re-keying scheme.[1]

Related works. In addition to the previously listed schemes that leverage the masking countermeasure, it was also proposed to use a leakage-resilient PRF for re-keying, as investigated in [MSJ12, BSH$^+$14, MSNF16, USS$^+$20]. Such a solution has been recently integrated in the ISAP Authenticated Encryption scheme [DEM$^+$17]. It does not rely on key-homomorphism and rather aims at limiting the manipulation of the long-term key in order to limit the attack vectors to Simple Power Analysis (SPA) attacks.

2 Background & definitions

2.1 Notations

We denote vectors with bold letters $\boldsymbol{v}$ and matrices with bold capital letters $\boldsymbol{M}$. We use the log notation for the logarithm in basis 2. For $n \in \mathbb{N}$, we denote by $[n]$ the set of integers from 1 to n and by $[0, n]$ the set of integers from 0 to n.

2.2 Physical model & LWPR

In order to define the LWPR problem, we need to define the physical model we are working with. In this respect, the main issue is that we must specify crypto-physical dark matter computations that mix mathematical operations and physical ones. For this purpose, we first observe that the elements of the vector $\boldsymbol{y} = \boldsymbol{K} \cdot \boldsymbol{r}$ are in $\mathbb{F}_p$. We then formalize as “physical rounding” the function modeling the side-channel information an adversary gets from noise-free leakages on this vector. The physical model we will consider for the rounding is a composition of two (more or less specialized) assumptions.

[1] Admittedly, the terms “compressive mapping” or “nonlinear filtering” would also be appropriate to describe the Hamming weight function. We use the term rounding because of its appealing analogy with the LWR problem that was already used in the fresh re-keying literature [DFH$^+$16].

One the one hand, we assume that the leaking device computes on binary-represented data: each value in $\mathbb{F}_p$ is therefore represented with (at least) $\lceil \log p \rceil$ bits. We denote as $\mathsf{g} : \mathbb{F}_p \to \{0,1\}^{\lceil \log p \rceil}$ the function associating to each element of $\mathbb{F}_p$ the binary representation of its representative in $[0, p-1]$. We also define $\mathsf{g}_m : \mathbb{F}_p^m \to \{0,1\}^{m\lceil \log p \rceil}$ as: $\mathsf{g}_m(\boldsymbol{v}) := \mathsf{g}(v_1)||\mathsf{g}(v_2)||\ldots||\mathsf{g}(v_m)$. We argue that this assumption is quite generic and captures the reality of most embedded computing devices deployed in current applications.

On the other hand, we need a more specialized assumption defining how the physical (noise-free) leakages depend on the $m\lceil \log p \rceil$ bits provided by g_m. This role will be played by the leakage function. We denote the leakage function computed on the binary representation of the manipulated data as $\mathsf{L}_\mathsf{g}(.)$ and use it as a parameter of our investigations.

We can then define a generic LWPR problem as follows.

Definition 1 (Learning with physical rounding)**.** Let $p, n, m \in \mathbb{N}^*$, p prime, for (unknown) $\boldsymbol{K} \in \mathbb{F}_p^{m\times(n+1)}$. The $\mathrm{LWPR}^{n,m}_{\mathsf{L}_\mathsf{g},p}$ sample distribution is given by:

$$\mathcal{D}_{\mathrm{LWPR}^{n,m}_{\mathsf{L}_\mathsf{g},p}} := \big(\boldsymbol{r}, \mathsf{L}_\mathsf{g}(\boldsymbol{K} \boxdot \boldsymbol{r})\big) \text{ for } \boldsymbol{r} \in \mathbb{F}_p^n \text{ uniformly random,}$$

where $\boldsymbol{K} \boxdot \boldsymbol{r} = \boldsymbol{K} \cdot (\boldsymbol{r}, 1)$ and $\mathsf{L}_\mathsf{g} : \mathbb{F}_p^m \to \mathbb{R}^d$ is the physical rounding function. Given query access to $\mathcal{D}_{\mathrm{LWPR}^{n,m}_{\mathsf{L}_\mathsf{g},p}}$ for a uniformly random $\boldsymbol{K}$, the $\mathrm{LWPR}^{n,m}_{\mathsf{L}_\mathsf{g},p}$ problem is (q, τ, μ, ϵ)-hard to solve if after the observation of q LWPR samples, no adversary can recover the key $\boldsymbol{K}$ with time complexity τ, memory complexity μ and probability higher than ϵ.

Note that $\boldsymbol{K}$ is multiplied with $(\boldsymbol{r}, 1)$ rather than $\boldsymbol{r}$. The additional $m\lceil \log p \rceil$-bit key addition is needed to obtain strong differential properties, as discussed in Section 4.2.1.

As already mentioned, as a starting point and as an interesting feasibility result, we will next consider the security that can be obtained with the Hamming weight function which the most frequently observed leakage model for standard CMOS devices [MOP07]. For this purpose, we first denote the Hamming weight function $\mathsf{HW}(\boldsymbol{v})$, defined on any vector $\boldsymbol{v}$ of length $t \in \mathbb{N}^*$ with coefficients in $\{0,1\}$ as $\mathsf{HW}(\boldsymbol{v}) = \sum_{i=1}^t \boldsymbol{v}_i$, where the sum is performed in $\mathbb{Z}$. We then consider two possible implementations of it:

- Parallel: $\mathsf{L}^p_\mathsf{g}(\boldsymbol{y}) : \boldsymbol{y} \mapsto \mathsf{HW}\big(\mathsf{g}_m(\boldsymbol{y})\big) = \mathsf{HW}\big(\mathsf{g}(\boldsymbol{y}_1)\big) + \mathsf{HW}\big(\mathsf{g}(\boldsymbol{y}_2)\big) + \ldots + \mathsf{HW}\big(\mathsf{g}(\boldsymbol{y}_m)\big)$.
- Serial: $\mathsf{L}^s_\mathsf{g}(\boldsymbol{y}) : \boldsymbol{y} \mapsto \Big(\mathsf{HW}\big(\mathsf{g}(\boldsymbol{y}_1)\big), \mathsf{HW}\big(\mathsf{g}(\boldsymbol{y}_2)\big), \ldots, \mathsf{HW}\big(\mathsf{g}(\boldsymbol{y}_m)\big)\Big)$.

The parallel case is reflecting a hardware implementation where the m vector elements are computed in parallel. The serial case is reflecting a software implementation where the m vector elements are computed one by one. (Intermediate levels of parallelism could be defined similarly). The serial case is a significantly more challenging context for securing implementations against side-channel attacks than the parallel case, since a b-bit Hamming weight function leaks approximately $\log(b)$ bits on average to the adversary.

Remark 1. Note that when $m = 1$, the $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_\mathsf{g},p}$ and $\mathrm{LWPR}^{n,m}_{\mathsf{L}^s_\mathsf{g},p}$ problems are identical. Note also that one instance of $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_\mathsf{g},p}$ generates the same amount of key material as m instances of $\mathrm{LWPR}^{n,1}_{\mathsf{L}^s_\mathsf{g},p}$, with the same amount of randomnesses. $\mathrm{LWPR}^{n,m}_{\mathsf{L}^s_\mathsf{g},p}$ samples can be easily converted into $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_\mathsf{g},p}$ samples (by summing them). Hence, if the $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_\mathsf{g},p}$ problem is not (q, τ, μ, ϵ)-hard, then the $\mathrm{LWPR}^{n,m}_{\mathsf{L}^s_\mathsf{g},p}$ is not $(q, \tau + q\theta(m), \mu, \epsilon)$-hard either, where $\theta(m)$ is the time complexity of adding m integers smaller than p.

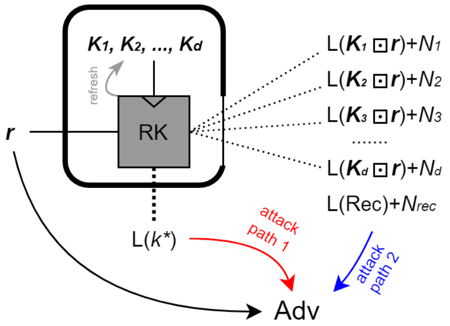


Figure 3: General adversarial model of our re-keying scheme.

2.3 A general adversarial model

While trying to break the LWPR assumption is one natural path to attack our re-keying scheme, it is not the only option. Side-channel security also relies on the fact that the re-keying function itself is well protected. Since crypto-physical dark matter computations are key-homomorphic, masking is a natural candidate for this purpose. It leads to the general adversarial model of Figure 3, where the first (red) attack path targets the leakage of the re-combined ephemeral key $\mathsf{L}(k^*)$ (i.e., the LWPR assumption) while the second (blue) attack path targets the noisy leakages of the shared computations $\boldsymbol{K}_1 \boxdot \boldsymbol{r}$, $\boldsymbol{K}_2 \boxdot \boldsymbol{r}$, …, $\boldsymbol{K}_d \boxdot \boldsymbol{r}$ together with the leakage generated by the shares' recombination.

We next formalize this adversarial model, starting with a definition of our re-keying scheme and following with the key recovery experiment it aims to keep hard.

Definition 2 (Re-keying scheme)**.** Let $n \in \mathbb{N}$ be a security parameter, and $d \in \mathbb{N}$ a number of shares. A re-keying scheme RK is made of the polytime algorithms:

- $\mathsf{Gen}(1^n, d)$. Generates the long-term key $\boldsymbol{K}$ and the initial sharing $\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K}_d$.
- $\mathsf{SharedMult}(\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K}_d, \boldsymbol{r})$. Generates the shares of the ephemeral key k^* (i.e., $k_1^*, k_2^*, \ldots, k_d^*$) from the d shares $\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots,$ $\boldsymbol{K}_d$ and a randomness $\boldsymbol{r}$.
- $\mathsf{Rec}(k_1^*, k_2^*, \ldots, k_d^*)$. Generates the ephemeral key k^* by recombining its shares.
- $\mathsf{Refresh}(\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K}_d)$. Replace the shares of the long-term key by fresh ones.

Definition 3 (Side-channel key recovery experiment $\mathsf{Exp}^{\mathsf{skr}}_{\mathcal{A},\mathsf{RK}}(n, d)$)**.** The experiment processes in three (setup, challenge and final) phases specified as:

- Setup phase. A long term key $\boldsymbol{K}$ is generated in function of the security parameter n and it is split into shares $\boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K}_d$ using the $\mathsf{Gen}(1^n, d)$ algorithm.
- Challenge phase. The adversary $\mathcal{A}$ performs q re-keying queries. For each query, the vector $\boldsymbol{r}$ is chosen at random, the $\mathsf{SharedMult}$ algorithm is computed on the shares of the long-term key and $\boldsymbol{r}$, and the shares of the long-term key are refreshed. The vector $\boldsymbol{r}$ is then given to $\mathcal{A}$ with the following leakages:
  - $\mathsf{L}(\boldsymbol{K}_i \boxdot \boldsymbol{r}) + N_i$, for $i \in [d]$ and some noise N_i – the shares' leakages.
  - $\mathsf{L}(\mathsf{Rec}) + N_{rec}$, for some noise N_{rec} – the leakage from the shares' recombination.
  - $\mathsf{L}(k^*)$, the noise-free leakage from the ephemeral key k^*.
- Final phase. The adversary $\mathcal{A}$ outputs a candidate for the long-term key k'. The output of the experiment is defined to be 1 if $k' = k$ and 0 otherwise.

We say that $\mathcal{A}$ succeeds, or breaks the re-keying scheme RK, with probability ϵ, time complexity τ, memory complexity μ and q queries if after q queries in a challenge phase bounded by these time and memory complexities, $\mathsf{Exp}^{\mathsf{skr}}_{\mathcal{A},\mathsf{RK}}(n,d) = 1$ with probability ϵ.

Remark 2. While the second attack path of Figure 3 (targeting the masked computations) is well investigated in the literature, the first attack path (targeting the LWPR samples) is new. We therefore start by studying this second attack path in Sections 3 and 4. For completeness, we also discuss the second attack path and how to choose the number of shares to reach a given security level in Section 5.3. Whether both attack paths can be combined in advanced attacks is an interesting scope for further research.

Remark 3. While the LWPR problem is stated for noise-free leakages (which is important since the ephemeral key is unshared), the secure implementation of masking generally requires a certain level of noise. Yet, contrary to the masking of nonlinear operations for which this necessary level of noise may increase with the number of shares [BCPZ16], the masking of a key-homomorphic primitive only requires a constant noise rate. Other advantages of the re-keying approach for masking are recalled in Section 5.3.

Remark 4. The adversarial model of Figure 3 does not explicitly show that the ephemeral key k^* is used in a (tweakable) block cipher in Figure 1. Concretely, the leakage $\mathsf{L}(k^*)$ therefore has to be understood as all the leakage that can be obtained on k^*. The starting assumption we study in this paper is the one of an adversary who does not obtain significantly more information than the Hamming weight of k^*. Analyzing more general classes of leakages is an important open problem, as will be discussed in Section 6.

2.4 Functions over $\mathbb{F}_p$ and vectorial Boolean functions

2.4.1 Cryptographic criteria & p-ary functions

We adapt tools from the analysis of cryptographic criteria for Boolean functions (from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ [CCH10a]) to study the cryptographic properties of functions from $\mathbb{F}_p^n$ to $\mathbb{F}_p$.

Definition 4 (p-ary function)**.** For p a prime, a p-ary function f in n variables (an n-variable p-ary function) is a function from $\mathbb{F}_p^n$ to $\mathbb{F}_p$. The set of all p-ary functions in n variables is denoted by $\mathcal{F}_{p,n}$, and $|\mathcal{F}_{p,n}| = p^{p^n}$. 2-ary functions are called Boolean.

Definition 5 (Algebraic normal form and algebraic degree (e.g., [Hou18]))**.** We call Algebraic Normal Form (ANF) of a p-ary function f its n-variable polynomial representation over $\mathbb{F}_p$ (i.e., belonging to $\mathbb{F}_p[x_1,\ldots,x_n]/(x_1^p - x_1,\ldots,x_n^p - x_n)$):

$$\mathsf{f}(x) = \sum_{S\subset[0,p-1]^n} a_S \left(\prod_{i\in[n]} x_i^{S_i}\right) = \sum_{S\subset[0,p-1]^n} a_S x^S,$$

where $a_S \in \mathbb{F}_p$. The ANF of f is unique, and the algebraic degree of f equals the global degree of its ANF: $\mathsf{deg}(\mathsf{f}) = \max_{\{S\,\mid\, a_S\neq 0\}} \sum_{i=1}^{n} S_i$ (with the convention that $\mathsf{deg}(0) = -\infty$). When $p = 2$, this can be expressed in terms of Hamming weight: $\mathsf{deg}(\mathsf{f}) = \max_{\{S\in\mathbb{F}_2^n\,\mid\, a_S\neq 0\}} \mathsf{HW}(S)$.

Definition 6 (Nonlinearity)**.** For $d \in \mathbb{N}^*$, the order-d nonlinearity $\mathsf{nl}_d(\mathsf{f})$ of a p-ary function $\mathsf{f} \in \mathcal{F}_{p,n}$, is the minimum Hamming distance between f and all the functions in $\mathcal{F}_{p,n}$ of degree at most d:

$$\mathsf{nl}_d(\mathsf{f}) = \min_{\mathsf{f}^*,\, \mathsf{deg}(\mathsf{f}^*)\leq d} \{d_H(\mathsf{f},\mathsf{f}^*)\},$$

where $d_H(\mathsf{f},\mathsf{f}^*)$ is the Hamming distance $|\{x \in \mathbb{F}_p^n \mid \mathsf{f}(x) \neq \mathsf{f}^*(x)\}|$ between f and f^*.

2.4.2 Cryptographic criteria & vectorial Boolean functions

We give definitions on vectorial Boolean functions that will be used in the paper [CCH10b], with criteria borrowed from the analysis of block ciphers that we will evaluate.

Definition 7 (Vectorial Boolean function)**.** A function from $\mathbb{F}_2^s$ to $\mathbb{F}_2^t$ is called vectorial Boolean function. For F an (s,t)-vectorial Boolean function, the t Boolean functions $\mathsf{f}_i : (x_1,\dots,x_s) \mapsto (\mathsf{F}(x_1,\dots,x_s))_i$, $1 \leq i \leq t$, with $(\mathsf{F}(x_1,\dots,x_s))_i$ the i-th bit of $\mathsf{F}(x_1,\dots,x_s)$ are called the coordinate functions of F. The 2^t-1 nonzero linear combinations of the coordinate functions are called component functions.

Definition 8 (Degree)**.** For a vectorial Boolean function F the algebraic degree $\mathsf{Deg}(\mathsf{F})$ is defined as the maximum of the algebraic degrees of its coordinate functions.

Definition 9 (MELP (e.g., [Vau99]))**.** Let a family of vectorial Boolean functions $(\mathsf{F}_K)_K$ from $\mathbb{F}_2^s$ to $\mathbb{F}_2^t$ be parameterised by a key K. Its Maximum Expected Linear Probability (MELP) is defined as:

$$\mathsf{MELP}((\mathsf{F}_K)_K) = \max_{a\in\mathbb{F}_2^s,\, b\in\mathbb{F}_2^t\setminus\{0\}} \frac{1}{|(\mathsf{F}_K)_K|} \sum_{\mathsf{F}_K\in(\mathsf{F}_K)_K} \left(\frac{\widehat{\mathsf{F}_K}(a,b)}{2^s}\right)^2 ,$$

with $\widehat{\mathsf{F}_K}(a,b) = \sum_{x\in\mathbb{F}_2^s} (-1)^{b\cdot\mathsf{F}_K(x)+a\cdot x}$ the coefficients of the Hadamard transform of F_K.

Definition 10 (MEDP (e.g., [Vau99]))**.** Let a family of vectorial Boolean functions $(\mathsf{F}_K)_K$ from $\mathbb{F}_2^s$ to $\mathbb{F}_2^t$ be parameterised by a key K. Its Maximum Expected Differential Probability is defined as:

$$\mathsf{MEDP}((\mathsf{F}_K)_K) = \max_{a\in\mathbb{F}_2^s\setminus\{0\},\, b\in\mathbb{F}_2^t} \frac{1}{|(\mathsf{F}_K)_K|} \sum_{\mathsf{F}_K\in(\mathsf{F}_K)_K} \frac{\delta_{\mathsf{F}_K}(a,b)}{2^s} ,$$

where $\delta_{\mathsf{F}_K}(a,b) = |\{x \in \mathbb{F}_2^s \mid \mathsf{F}_K(x) \oplus \mathsf{F}_K(x\oplus a) = b\}|$ is the number of solutions of the differential equation defined by the mask (a,b), with $\oplus$ the bitwise XOR.

Definition 11 (ε-AXU [CW79])**.** A family of keyed functions $\mathsf{H} : A \to B$ is ε-almost XOR universal if:

$$\forall x \neq x' \in A,\ \forall d \in B,\ |\{h_K \in \mathsf{H}, \mathsf{h}_K(x) \oplus \mathsf{h}_K(x') = d\}| \leq \varepsilon|\mathsf{H}|$$

In particular, for $(\mathsf{F}_K)_K$ an ε-AXU family of keyed functions from $\mathbb{F}_2^s$ into $\mathbb{F}_2^t$, we have:

$$\mathsf{MEDP}((\mathsf{F}_K)_K) = \frac{1}{2^s} \max_{a\neq 0,b} \frac{1}{|(\mathsf{F}_K)_K|} \sum_{x\in\mathbb{F}_2^s} |\{\mathsf{F}_K \in (\mathsf{F}_K)_K, \mathsf{F}_K(x) \oplus \mathsf{F}_K(x\oplus a) = b\}| \leq \varepsilon ,$$

since $|\{\mathsf{F}_K \in (\mathsf{F}_K)_K, \mathsf{F}_K(x) \oplus \mathsf{F}_K(x\oplus a) = b\}| \leq \varepsilon|(\mathsf{F}_K)_K|$.

3 Negative results: Small p or small n are not enough

The operations involved while computing a LWPR sample can be separated in two parts: on the one hand a matrix-vector multiplication over $\mathbb{F}_p$, on the other hand the physical rounding function. The first part is a linear operation over $\mathbb{F}_p$. Therefore, if the second part has a cryptographic weakness in characteristic p, it could be used to break the LWPR problem. As a warm-up, we show the LWPR problem cannot be hard with small p or n values. For this purpose, we first show that the $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},3}$ problem is no harder than solving a quadratic system of equations in characteristic 3. We then give a minimum (necessary) condition on the vector size n for the LWPR problem to be hard.

3.1 $\mathbb{F}_3$ is not secure enough

We first focus on the function $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y})$ when $\boldsymbol{y}$ is a single element of $\mathbb{F}_3$. In this case, the adversary observes the real $\mathsf{L}^p_{\mathsf{g}}(y)$ which belongs to $\mathbb{Z}$ and can be embedded in $\mathbb{F}_3$. That is, she observes the outputs of the 3-ary (ternary) function $\mathsf{f}(y) = \mathsf{HW}(\mathsf{g}(y)) \mod 3$ associating to each $y \in \mathbb{F}_3$ the Hamming weight of its binary representation.

This function has algebraic degree at most 2 since it is a ternary function in one variable, and it can be determined by solving a linear system over $\mathbb{F}_3$. More specifically, since f satisfies $\mathsf{f}(0) = \mathsf{HW}(\mathsf{g}(0)) = \mathsf{HW}((0,0)) = 0$, $\mathsf{f}(1) = \mathsf{HW}(\mathsf{g}(1)) = \mathsf{HW}((0,1)) = 1$, and $\mathsf{f}(2) = \mathsf{HW}(\mathsf{g}(2)) = \mathsf{HW}((1,0)) = 1$, we directly obtain $\mathsf{f}(y) = y^2$.

Next, coming back to the general case, we focus on the function $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y})$ where $\boldsymbol{y} \in \mathbb{F}_3^m$. When the adversary embeds $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y})$ in $\mathbb{F}_3$, it corresponds to the ternary function $\mathsf{f}'(\boldsymbol{y}) = \mathsf{HW}(\mathsf{g}_m(\boldsymbol{y})) \mod 3$. In other words, f' is simply the direct sum of m times the previous function f, and its algebraic degree is equal to the one of f. In particular: $\mathsf{f}'(\boldsymbol{y}) = \sum_{i=1}^m \mathsf{f}(\boldsymbol{y}_i) = \sum_{i=1}^m \boldsymbol{y}_i^2$, and f' has algebraic degree 2 for $p = 3$.

Finally, since each $\boldsymbol{y}_i$ is the result of a product between a row of $\boldsymbol{K}$ and $(\boldsymbol{r}, 1)$ over $\mathbb{F}_3$, the adversary can extract from each $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},3}$ sample a quadratic relation over $\mathbb{F}_3$ in the elements of $\boldsymbol{K}$, namely:

$$\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{K} \boxdot \boldsymbol{r}) \mod 3 = \sum_{i=1}^m \left(\sum_{j=1}^n \boldsymbol{K}_{i,j}\boldsymbol{r}_j + \boldsymbol{K}_{i,n+1} \right)^2 \mod 3.$$

By collecting LWPR samples the adversary can linearize this quadratic system in at most $m((n+1)(n+2))/2$ unknowns, and determine a valid key for these samples by solving it. Hence, considering that solving a linear system through standard Gaussian elimination has at most a cubic complexity, we conclude that $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},3}$ is not $(\mathcal{O}(mn^2), \mathcal{O}(m^3n^6), \mathcal{O}(m^3n^6), 1)$-hard, making such an instance hardly useful for practical applications.[2] This attack can be extended for all prime p's. But since its complexity increases exponentially with p, it will not be a security issue for p big enough:

Proposition 1. *Let $n, m, p \in \mathbb{N}$, p a prime, solving the $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},p}$ problem can be reduced to solving an algebraic system of degree $p-1$ in characteristic p.*

Proof. We begin by considering a function $\mathsf{f} \in \mathcal{F}_{p,1}$, the p-ary function defined for $y \in \mathbb{F}_p$ as $\mathsf{f}(y) = \mathsf{HW}(\mathsf{g}(y)) \mod p$. Note that independently of the exact expression of the function g, the HW function gives an element in $\mathbb{Z}$, and considering its remainder modulus p always gives a function from $\mathbb{F}_p$ to $\mathbb{F}_p$. Since all functions from $\mathbb{F}_p$ to $\mathbb{F}_p$ are the functions of $\mathcal{F}_{p,1}$, the degree of f is at most $p-1$ (see Definition 5). Then, we focus on the p-ary function $\mathsf{f}' : \mathbb{F}_p^m \mapsto \mathbb{F}_p$ defined as $\boldsymbol{y} \to \mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y}) \mod p$. Note that:

$$f'(\boldsymbol{y}) = \left(\sum_{i=1}^m \mathsf{HW}(\mathsf{g}(\boldsymbol{y}_i)) \right) \mod p = \sum_{i=1}^m \mathsf{HW}(\mathsf{g}(\boldsymbol{y}_i)) \mod p = \sum_{i=1}^m f(\boldsymbol{y}_i).$$

Hence $\mathsf{deg}(\mathsf{f}') = \mathsf{deg}(\mathsf{f})$ and since $\mathsf{deg}(\mathsf{f}) \leq p-1$, we obtain $\mathsf{deg}(\mathsf{f}') \leq p-1$. Since each $\boldsymbol{y}_i$ is the result of the linear combination over $\mathbb{F}_p$ of the i-th row of $\boldsymbol{K}$ and the public vector $\boldsymbol{r}$, each $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},p}$ sample leads to a degree at most $p-1$ equation in the key elements, in characteristic p. More precisely: $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{K} \boxdot \boldsymbol{r}) \mod p = \mathsf{f}'(\sum_{i=1}^n \boldsymbol{K}_{1,i}\boldsymbol{r}_i + \boldsymbol{K}_{1,n+1}, \ldots, \sum_{i=1}^n \boldsymbol{K}_{m,i}\boldsymbol{r}_i + \boldsymbol{K}_{m,n+1})$. Therefore, an adversary solving the algebraic system given by the different $\boldsymbol{r}$ values recovers $\boldsymbol{K}$ and breaks the $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},p}$ problem. □

[2] The system may give various solutions and not only the correct $\boldsymbol{K}$. We ignore this issue since our goal in this section is only to show that small p's cannot guarantee higher security.

3.2 Small vectors are not secure enough

Contrary to the previous result which holds for any m (and shows that crypto-physical dark matter cannot be secure with small p's even if implemented in parallel), we now consider an attack against $\mathrm{LWPR}^{n,1}_{\mathsf{L}_\mathsf{g},p}$ taking advantage of a small value of n, which only imposes a condition for serial implementations. Roughly, when $m = 1$, each sample gives the Hamming weight of a known linear combination of the key elements. We show next how this information over different samples can lead to an attack, provided n is small.

First, note that each sample of $\mathrm{LWPR}^{n,1}_{\mathsf{L}_\mathsf{g},p}$ has the form $(\boldsymbol{r}, u)$ with $u = \mathsf{HW}(\mathsf{g}(\sum_{i=1}^{n} \boldsymbol{k}_i \boldsymbol{r}_i + \boldsymbol{k}_{n+1}))$ since the key is reduced to a vector when $m = 1$. The value u belongs to $\mathbb{Z}$, and considering the binary decomposition g over ℓ bits, $0 \leq u \leq \ell$, the function $\mathsf{HW}(\mathsf{g}(\cdot))$ is surjective over $[0, \ell]$ and $|\mathsf{HW}(\mathsf{g}(\cdot))^{-1}(u)|$ takes different values for $u \in [0, \ell]$.

Let us denote as $\mathcal{A}_u$ the set of preimages of u through $\mathsf{HW}(\mathsf{g}(\cdot))$. Then, each sample gives the information $\sum_{i=1}^{n} \boldsymbol{k}_i \boldsymbol{r}_i + \boldsymbol{k}_{n+1} \in \mathcal{A}_u$. So taking $a \in \mathcal{A}_u$, there exists one of the $|\mathcal{A}_u|$ equations $\sum_{i=1}^{n} \boldsymbol{k}_i \boldsymbol{r}_i + \boldsymbol{k}_{n+1} = a$ which is the correct one. After collecting $n+1$ samples, and the corresponding $u_1, \ldots, u_n$ values, one of the $\prod_{i=1}^{n+1} |\mathcal{A}_{u_i}|$ linear systems in $n+1$ unknowns over $\mathbb{F}_p$ characterizes the key. We next show that a few extra samples are sufficient in order to verify if a candidate key is the right key. Consequently the $\mathrm{LWPR}^{n,1}_{\mathsf{L}_\mathsf{g},p}$ problem can be solved by solving a certain amount of linear systems. We additionally highlight how this amount evolves with n, and how to reduce it.

The inner product between $(\boldsymbol{r}, 1)$ and $\boldsymbol{k}$ is uniformly distributed in $\mathbb{F}_p$ (any non null element modulus p generates the multiplicative group). Therefore the probability for a wrong key $\boldsymbol{k}'$ to give the same u is $|\mathcal{A}_u|/p$. Let us denote $M = \max_{u \in [0,\ell]} |\mathcal{A}_u|$. Then, the probability of having a wrong key consistent with t samples is lower than or equal to $(M/p)^t$. Considering $M/p \leq 1/2$ with λ samples (where λ is the bit-security parameter) already ensures that $\boldsymbol{k}'$ is consistent with the samples only with a negligible probability. More precisely, for $2^{\ell-1} < p < 2^{\ell}$ where $\ell \in \mathbb{N}^*$ and g corresponding to the usual binary representation over ℓ bits, we get $M/p \leq \binom{\ell}{\lceil \ell/2 \rceil}/2^{\ell-1}$, which allows us to determine the number of extra samples to consider for rejecting the wrong keys.

The amount of linear systems over $\mathbb{F}_p$ to solve can be as high as M^{n+1}, which is at most $\binom{\ell}{\lceil \ell/2 \rceil}^{n+1}$ for the definition of g we consider. For each linear system, the attack consists in solving the system given by the first $n+1$ samples, which can be done in time $\mathcal{O}(n^3)$ with standard Gaussian elimination, and in testing if the obtained key is consistent with $t \leq \lambda$ extra samples, which costs t inner products, evaluations of g and HW. Assuming the cost of the t evaluations is smaller than $\mathcal{O}(n^3)$ (the cost of solving the linear system), the attack cost in time is $\mathcal{O}(M^{n+1} n^3)$ and it requires at most $n+1+\lambda$ samples. Taking p as an upper bound on M gives attacks when $(n+1)\log p + 3 \log n < \lambda$. For mid-size p values that are interesting for implementation purposes (e.g., $p \approx 2^8$, 2^{16}, 2^{32}), this inequality therefore sets a condition for the minimum vector size n. If not respected, we can conclude that the resulting $\mathrm{LWPR}^{n,1}_{\mathsf{L}_\mathsf{g},p}$ problem is not $(\mathcal{O}(n+1+\lambda), \mathcal{O}(p^{n+1} n^3), \mathcal{O}(n^3), 1 - \mathsf{negl}(\lambda))$-hard.

This attack exists for all values of n, but its complexity increases exponentially with n. In the following we show that even when generalizing the attack, the complexity still increases exponentially with n, and also with m for $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_\mathsf{g},p}$.

Generalization. Having M^{n+1} systems to solve is the worst case in the previously described attack. Instead, an adversary can take more favorable samples, the ones such that the values of $|\mathcal{A}_u|$ are smaller. The values of $|\mathcal{A}_u|$ for u close to 0 or to ℓ are smaller than for medium Hamming weight, namely $|\mathcal{A}_u| \leq \binom{\ell}{u}$. Let us denote the set $S_v = [0, v] \cup [\ell - v, \ell]$ and $M_v = 2\sum_{i=0}^{v} \binom{\ell}{i}$. Collecting only the samples such that $u \in S_v$, the adversary reduces the amount of linear systems to solve from M^{n+1} to at most M_v^{n+1}. The probability of getting $n+1$ samples following this constraint can be determined with a binomial law depending on v and the number of samples (it is not with probability 1 as in the simpler case). Indeed, since $\boldsymbol{r}$ is uniformly distributed, the probability P_k of having $u \in S_v$ is $P_v = \left(\sum_{u \in S_v} |\mathcal{A}_u|\right)/p$. Then, the probability of having at least $n+1$ over q samples satisfying this constraint is given by $1 - F(n, P_v, q)$ where F denotes the cumulative distribution function of the binomial law with success parameter P_v and number of tries q. The time cost of the attack is decreased to $\mathcal{O}(M_v^{n+1} n^3)$, but the probability of success is reduced (varying with the number of samples q and the choice of v). Finally, the complexity of this modified attack remains exponential in n, showing that it applies for small values of n and not after.

Note that when m increases, this attack on $\mathrm{LWPR}^{n,m}_{\mathsf{L}^p_{\mathsf{g}},p}$ also becomes impractical. In this case, a sample has the shape $(\boldsymbol{r}, u)$, where $u = \sum_{i=1}^{m} u_i$ is in $\mathbb{Z}$ with each u_i obtained as $u_i = \mathsf{HW}(\mathsf{g}(\sum_{j=1}^{n} \boldsymbol{K}_{i,j}\boldsymbol{r}_j + \boldsymbol{K}_{i,n+1}))$. Hence $0 \le u \le \ell m$. We denote $\mathcal{A}_u^m$ the set of preimages of u through $\mathsf{HW}(\mathsf{g}_m(\cdot))$, the size of $|\mathcal{A}_u^m|$ is growing exponentially in m. Indeed, writing $|\mathcal{A}_u^m|$ in terms of $|\mathcal{A}_u|$ we get:

$$\forall u \in [0, m\ell], \quad |\mathcal{A}_u^m| = \sum_{\substack{u_1,\ldots,u_m \in [0,\ell]^m \\ u_1+\cdots+u_m = u}} \prod_{i=1}^{m} |\mathcal{A}_{u_i}|.$$

As for the case $m = 1$, the adversary can take advantage of samples where u is close to 0 or $m\ell$, but the probability of getting such samples is decreasing exponentially in m.

4 First analysis and proposed instance

We now move to the analysis of crypto-physical dark matter instances that can lead to hard LWPR problems. As a first step in this direction, we put forward desirable security properties that can be used in order to rule out a number of standard attacks. As already mentioned in introduction, this analysis is admittedly not exhaustive: it is only proposed to support our claim that re-keying in the noise-free model of Figure 2(b) can provide stronger security guarantees than the initial proposal of Medwed et al. [MSGR10], which is only secure in the noisy model of Figure 2(a). We proceed in two steps for this purpose. First, we extend the negative results of the previous section which analyze our construction in $\mathbb{F}_p$. We show that with sufficiently large p and n values, we can lower bound the algebraic degree and the nonlineariy of the function generating LWPR samples. Next, we complement this analysis with an evaluation in $\mathbb{F}_2$. In this case, we focus on the cryptographic properties of $\boldsymbol{K} \boxdot \boldsymbol{r}$ when interpreted as a function over binary fields, and focus on its differential/linear properties and its algebraic degree.

4.1 Analysis in characteristic p

As mentioned in Section 3, the product $\boldsymbol{K} \boxdot \boldsymbol{r}$ is linear over $\mathbb{F}_p$. Hence, our focus is on the (linear invariant) cryptographic criteria of the remaining function L_g considered over $\mathbb{F}_p$, denoted as f'.[3] In Section 3.1, we saw an upper bound on the degree of such a function. We now prove a lower bound on this degree. It allows us to thwart attacks based on solving a linearized algebraic system. Thanks to this bound on the degree, we also derive a bound on the nonlinearity of small order of the function f'. It enables us to prevent attacks based on solving low-degree noisy algebraic systems, relying on a good approximation of f' by a low degree function. We first introduce the iterated Hamming weight function, similarly to the iterated logarithm, that will be used to prove the lower bound on the degree.

[3] Criteria over $\mathbb{F}_p^i$ could be considered too, but since the Hamming weight function only exceeds p when $m \ge \frac{p}{\log p}$, such a generalization will not lead to relevant observations for our intended instances.

Definition 12 (Iterated Hamming weight function)**.** Let $n \in \mathbb{N}$, we define the iterated Hamming weight as:

$$\mathsf{it_H}(n) = \begin{cases} 0 & \text{if } n \leq 1, \\ 1 + \mathsf{it_H}\left(\max\limits_{2 \leq x \leq n} \mathsf{HW}(\mathsf{g}(x))\right) & \text{if } n > 1. \end{cases}$$

This definition allows us to prove the following two results:

Proposition 2 (Degree lower bound)**.** *Let $m, p \in \mathbb{N}^*$, p odd prime, and f' the p-ary function defined as $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y}) \mod p$, then $\mathsf{deg}(\mathsf{f}') \geq (p-1)^{\frac{1}{\mathsf{it_H}(p-1)}}$.*

Proof. Let us consider f the p-ary function in one variable which associates to each element $y \in \mathbb{F}_p$ the element in $\mathbb{F}_p$ corresponding to $\mathsf{HW}(\mathsf{g}(y))$. We show that applying f iteratively $\mathsf{it_H}(p-1)$ times gives the function y^{p-1}: the one associating 0 to 0 and 1 to any nonzero element. Note that $f(0) = \mathsf{HW}(\mathsf{g}(0)) = 0$, $f(1) = \mathsf{HW}(\mathsf{g}(1)) = 1$ and (considering the order in $\mathbb{Z}$) for $x \geq 2$, we have $1 \leq \mathsf{f}(y) < y$. Then, for each $y \neq 0$ there is a number of iteration $s \in \mathbb{N}^*$ such that for all $t \in \mathbb{N}^*$, $t \geq n$, $\mathsf{f}^{\circ t}(y) = \mathsf{f}(\mathsf{f}(\cdots(\mathsf{f}(y))\cdots)) = 1$, where we denote $\mathsf{f}^{\circ t}$ the function consisting in iterating t times f.

By construction the function $\mathsf{it_H}(\cdot)$ is non decreasing. Hence, for all $x \in [p-1]$ we have $\mathsf{it_H}(x) \leq \mathsf{it_H}(p-1)$ and for all $y \in \mathbb{F}_p^*$ we obtain $\mathsf{f}^{\circ \mathsf{it_H}(p-1)}(y) = 1$. Since $\mathsf{f}^{\circ \mathsf{it_H}(p-1)}(0) = 0$ we can conclude that $\mathsf{f}^{\circ \mathsf{it_H}(p-1)}$ and y^{p-1} take the same values over the whole $\mathbb{F}_p$, hence it is the same p-ary function. Due to the uniqueness of the ANF we use that $\mathsf{deg}(\mathsf{f}^{\circ \mathsf{it_H}(p-1)}) = p-1$, and therefore $\mathsf{deg}(\mathsf{f}) \geq (p-1)^{\frac{1}{\mathsf{it_H}(p-1)}}$ (since for $s \in \mathbb{N}^*, \mathsf{deg}(f^{\circ s}) \leq \mathsf{deg}(\mathsf{f})^s$).

Eventually, the function f' is the direct sum of m times the function f. Therefore $\mathsf{deg}(\mathsf{f}') = \mathsf{deg}(\mathsf{f})$, which allows us to conclude. □

Proposition 3 (Order-d nonlinearity lower bound)**.** *Let $m, p \in \mathbb{N}^*$, p odd prime, and f' the p-ary function defined as $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y}) \mod p$, then $\forall d \in [\lceil (p-1)^{\frac{1}{\mathsf{it_H}(p-1)}} \rceil - 1]$, $\mathsf{nl}_d(\mathsf{f}') \geq p^{m-1}$.*

Proof. We use a result on the p-ary Reed Muller codes. The p-ary Reed Muller code of order d and length p^n denoted $RM_p(d,m)$ consists in all the p^m tuples corresponding to $(h(\boldsymbol{y}))_{\boldsymbol{y} \in \mathbb{F}_p^m}$ where h is a m-variable p-ary function of degree at most d. Abusing notations, we note $h \in RM_p(d,m)$. For all $d \in \mathbb{N}$ such that $d < p$, the minimal Hamming distance of such code is $(p-d)p^{m-1}$ (e.g., [PW04], page 3).

Since $\mathsf{deg}(\mathsf{f}') \leq p-1$, $\mathsf{f}' \in RM_p(p-1, m)$, and therefore for all $h \in RM_p(p-1, m)$ such that $h \neq \mathsf{f}'$ we have $d_H(\mathsf{f}, h) \geq p^{m-1}$. Using Lemma 2 for all $d \leq \lceil (p-1)^{\frac{1}{\mathsf{it_H}(p-1)}} \rceil - 1$ we know that $\mathsf{f}' \notin RM_p(d,m)$, therefore f' is at Hamming distance at least p^{m-1} from all functions of degree at most d, which allows to conclude. □

Finally, we can also prove a better bound for the first-order nonlinearity:

Proposition 4 (First-order nonlinearity lower bound)**.** *Let $m, p \in \mathbb{N}^*$, p odd prime, $\ell = \lceil \log(p-1) \rceil$, and f' the p-ary function defined as $\mathsf{L}^p_{\mathsf{g}}(\boldsymbol{y}) \mod p$, if $m\ell < p-1$ then $\mathsf{nl}_1(\mathsf{f}') \geq p^m - \max\left(p^{m-1}(m\ell+1), \binom{\ell m}{\lceil \ell m/2 \rceil}\right)$.*

Proof. The first-order nonlinearity gives the minimum Hamming distance to constant functions and degree one functions. We study these two cases separately for this proof.

All degree one m-variable functions $\mathsf{l}_{\boldsymbol{a}}(\boldsymbol{y})$ have an ANF of the following shape: $\boldsymbol{a}_0 + \sum_{i=1}^m \boldsymbol{a}_i \boldsymbol{y}_i$, where at least one of the $\boldsymbol{a}_i$ is not null for $i \in [m]$. Without loss of generality, let us consider $\boldsymbol{a}_m \neq 0$. Then, for any fixed $\boldsymbol{z} \in \mathbb{F}_p^{m-1}$, the function $\mathsf{l}_{\boldsymbol{a}}(\boldsymbol{z}, \boldsymbol{y}_m)$ is a bijection of $\mathbb{F}_p$ since $\mathbb{F}_p$ is a prime field. It implies that $\mathsf{l}_{\boldsymbol{a}}$ takes each value of $\mathbb{F}_p$ exactly p^{m-1} times. The function f' has values only in the range $[0, m\ell]$, and since $m\ell < p-1$, all the elements in $[m\ell+1, p-1]$ have no preimage through f'. It allows deriving the following bound on the distance between f' and any degree 1 function: $\mathsf{HD}(\mathsf{l}_{\boldsymbol{a}}, \mathsf{f}') \geq p^{m-1}(p-m\ell-1)$.

The closest constant function to f' in Hamming distance is the one equal to the value of $\mathbb{F}_p$ taken the most by f'. Since $m\ell < p-1$, the value of f' seen as an integer in $[0, p-1]$ is equal to $\mathsf{HW}(\mathsf{g}(\boldsymbol{y}_1), \dots, \mathsf{g}(\boldsymbol{y}_m))$ considered over $\mathbb{Z}$. The binary vector $(\mathsf{g}(\boldsymbol{y}_1), \dots, \mathsf{g}(\boldsymbol{y}_m))$ has length ℓm, and the maximal number of length-ℓm binary vector having the same Hamming weight is given by the central binomial coefficient $\binom{\ell m}{\lceil \ell m/2 \rceil}$. Hence, the value taken the most by f' is taken at most $\binom{\ell m}{\lceil \ell m/2 \rceil}$ times, which gives the following distance to any constant function $c(\boldsymbol{y})$: $\mathsf{HD}(c, \mathsf{f}') \geq p^m - \binom{\ell m}{\lceil \ell m/2 \rceil}$.

We finally conclude from the two parts that the following holds:

$$\begin{aligned}\mathsf{nl}_1(\mathsf{f}') &= \min\left(\min_{\mathsf{f}^*,\, \mathsf{deg}(\mathsf{f}^*)=1} \{\mathsf{HD}(\mathsf{f}', \mathsf{f}^*)\}, \min_{\mathsf{f}^*,\, \mathsf{deg}(\mathsf{f}^*)\leq 0} \{\mathsf{HD}(\mathsf{f}, \mathsf{f}^*)\}\right), \\ &\geq \min\left(p^{m-1}(p-m\ell-1), p^m - \binom{\ell m}{\lceil \ell m/2 \rceil}\right), \\ &\geq p^m - \max\left(p^{m-1}(m\ell+1), \binom{\ell m}{\lceil \ell m/2 \rceil}\right). \qquad \square\end{aligned}$$

Discussion. Various techniques can be used to solve a noisy algebraic system of fixed degree. We use the higher-order correlation approach of [Cou02] to encompass different attacks and derive the corresponding complexities. Higher-order correlation attacks consists in approximating f' by a degree d function h, and in solving systems of equations until one is such that f' and h coincide on all these equations. In the system of equations given by the output of f', only the key elements are unknown. Therefore, solving the correct algebraic system allows retrieving the key. Following, the time complexity of solving a noisy degree d system of equation over $\mathbb{F}_p$ can be written as $C(1-\varepsilon)^{-D}$ with:

- C the time complexity to solve a degree d system of equations in V variables over $\mathbb{F}_p$,
- $(1-\varepsilon)$ the probability of the approximation to be correct for one equation,
- V is the number of variables (i.e., at least the number of key variables k, but it can be more if techniques introducing new variables –such as linearization– are used).
- D the quantity of data necessary (at least the number of variables V),

For illustration, we consider the complexity of linearizing the degree d system and solving the linear system obtained. It allows deriving concrete estimations of the complexity for three different attacks. For this purpose, we first observe that the number of variables after linearization is $V_d = |\{\boldsymbol{v} \in [0, p-1]^k,\ 1 \leq \sum_{i=1}^{k} \boldsymbol{v}_i \leq d\}|$ and $C = \mathcal{O}((V_d)^\omega)$, where ω is the exponent in the complexity of Gaussian elimination.

In the first case, the adversary aims to solve an exact algebraic system. It corresponds to $\varepsilon = 0$ and $d = \mathsf{deg}(\mathsf{f}')$. The time complexity is then $\mathcal{O}((V_{\mathsf{deg}(\mathsf{f}')})^\omega)$ and the bound on the degree from Proposition 2 allows us to conclude. The second attack targets a noisy linear system corresponding to $\varepsilon = \mathsf{nl}_1(\mathsf{f}')/p^m$ and $d = 1$. In this case, $V_1 = k$ and the time complexity is at least $\mathcal{O}(k^\omega(p^m/(p^m - \mathsf{nl}_1(\mathsf{f}')))^k)$. Hence, the bound of Proposition 4 provides the required estimation. Finally, if the adversary targets a noisy system of higher algebraic degree $d > 1$, we have $\varepsilon = \mathsf{nl}_d(\mathsf{f}')/p^m$. The time complexity is then at least $\mathcal{O}(V_d^\omega(p^m/(p^m - \mathsf{nl}_d(\mathsf{f}')))^{V_d})$ and the bound of Proposition 3 allows us to conclude.

Those results illustrate that the proposed crypto-physical dark matter operations lead to LWPR samples that resist some standard cryptanalysis techniques. We note that the attacks outlined are not claimed to be optimal. For example, using Gröbner basis algorithms such as F4 [Fau99] should improve over the aforementioned linearization techniques, which we leave as an interesting scope for further investigations.

4.2 Analysis in characteristic 2

We now consider a complementary analysis of the crypto-physical dark matter function $\mathsf{f} = \mathsf{L_g}(\boldsymbol{K} \boxdot \boldsymbol{r})$ interpreted over binary fields. In contrast with the previous results in $\mathbb{F}_p$ where the security of the related LWPR problem mostly depends on the Hamming weight function, such a leakage function is actually weak over binary fields (e.g., $\mathsf{L_g}(.) \bmod 2$ is a linear relation). Therefore, we rather rely on $\mathsf{h}_{\boldsymbol{K}} = \boldsymbol{K} \boxdot \boldsymbol{r}$ in $\mathbb{F}_p$ to provide security in the binary case. Our main result in this direction is to show that the MEDP (Maximum Expected Differential Probability) of $\mathsf{h}_{\boldsymbol{K}} = \boldsymbol{K} \boxdot \boldsymbol{r}$ can be bounded by leveraging existing results on universal hash functions. We complement this result by heuristic investigations on small instances from which we conclude that its MELP (Maximum Expected Linear Probability) follows a similar trend, and that its algebraic degree is close to maximal.

For this purpose, we first define the vectorial Boolean functions we will study. Denote ℓ the smallest integer such that $2^\ell > p$, that is the size in bits of the representation of an element of $\mathbb{F}_p$. Denote $\boldsymbol{x}_{[2]} \in \mathbb{F}_{2^\ell}^i$ a representation of the vector $\boldsymbol{x}_{[p]} \in \mathbb{F}_p^i$ over a binary field, obtained by representing each element of $\boldsymbol{x}$ in $\mathbb{F}_{2^\ell}$. We consider the family of functions $\mathsf{h}'_{\boldsymbol{K}} : \mathbb{F}_{2^\ell}^n \to \mathbb{F}_{2^\ell}^m$ defined for each key $\boldsymbol{K} \in \mathbb{F}_p^{m\times(n+1)}$ by $\mathsf{h}'_{\boldsymbol{K}}(\boldsymbol{r}_{[2]}) = (\boldsymbol{K}_{1\ldots n} \cdot \boldsymbol{r}_{[p]} + \boldsymbol{K}_{n+1})_{[2]}$, where $\boldsymbol{K}_{1\ldots n}$ is the matrix made of the n first columns of $\boldsymbol{K}$.

4.2.1 Differential analysis

We argue about the security of our construction by exhibiting its MEDP. We note that this security property is commonly studied for block ciphers and message authentication codes, but it is usually hard to obtain a good estimate of it without constraining assumptions. Here, we get the exact MEDP of the construction. The very first universal family of hash functions, called H_1, introduced by Wegman and Carter in [CW79], allows us to derive this result for our construction. It works in two steps: (1) we show that h' with $m = 1$ is $\frac{\alpha}{p}$-almost XOR universal with α close to 1; (2) we use tweaks of classical results on concatenation to show that h' is $\left(\frac{\alpha}{p}\right)^m$ XOR universal. Concretely, our construction h' with $m = 1$ is a modification of H_1 where the multiplication over $\mathbb{F}_p$ is replaced with a scalar product over $\mathbb{F}_p$. Denote $x_{[2]}$ the natural decomposition of $x \in \mathbb{F}_p$ over $\mathbb{F}_2^\ell$, with $2^\ell > p$ (i.e., seeing x in $\mathbb{N}$ and using its decomposition in basis 2). It yields:

Proposition 5 (($\boldsymbol{k} \boxdot \boldsymbol{r})_{[2]}$ is AXU)**.** *Let $\boldsymbol{k}$ be an element of $(\mathbb{F}_p^n \setminus \{0\}) \times \mathbb{F}_p$. Define $\mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{i=1}^n \boldsymbol{k}_i \boldsymbol{r}_i + \boldsymbol{k}_{n+1}$ mod p. Then, the family $(\mathsf{h}'_{\boldsymbol{k}})_{\boldsymbol{k}}$ defined for all $\boldsymbol{k}$ as $\mathsf{h}'_{\boldsymbol{k}}(\boldsymbol{r}) = \mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r})_{[2]}$ from $\mathbb{F}_p^n$ into $\mathbb{F}_2^\ell$, with $2^\ell \geq p$, is $\frac{\alpha}{p}$-almost XOR universal, with $\alpha = \frac{2^\ell}{p}\left(1 + \frac{1}{p^n - 1}\right)$.*

Proof. For $x \mapsto x_{[2]}$ a function from $\mathbb{F}_p$ to $\mathbb{F}_2^\ell$, consider β such that for any $i \in \mathbb{F}_2^\ell$, $|\{a \in \mathbb{F}_p \,|\, a_{[2]} = i\}| \leq \beta$. Let K a random variable over the universe of keys $\Omega = (\mathbb{F}_p^n \setminus \{0\}) \times \mathbb{F}_p$, which has cardinality $(p^n - 1)p$. Let $\boldsymbol{r}, \boldsymbol{r}' \in \mathbb{F}_p^n$, $\boldsymbol{r} \neq \boldsymbol{r}'$. Denote $\boldsymbol{k} \in \Omega = (\boldsymbol{k}_1, \ldots, \boldsymbol{k}_{n+1})$, $\mathsf{h}_{\boldsymbol{k}} : \mathbb{F}_p^n \to \mathbb{F}_p$, $\mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r}) = \sum_{i=1}^n \boldsymbol{k}_i \boldsymbol{r}_i + \boldsymbol{k}_{n+1}$. Then for any $d \in \mathbb{F}_2^\ell$:

$$
\begin{aligned}
& \Pr(\mathsf{h}_K(\boldsymbol{r})_{[2]} \oplus \mathsf{h}_K(\boldsymbol{r}')_{[2]} = d), \\
= & \sum_{u \in \mathbb{F}_2^\ell} \Pr((\mathsf{h}_K(\boldsymbol{r})_{[2]},\, \mathsf{h}_K(\boldsymbol{r}')_{[2]}) = (u,\, u \oplus d)), \\
= & \sum_{u \in \mathbb{F}_2^\ell} \sum_{(a,b) \in \mathbb{F}_p^2} \Pr((\mathsf{h}_K(\boldsymbol{r}),\, \mathsf{h}_K(\boldsymbol{r}')) = (a,\, b)) \Pr((a_{[2]},\, b_{[2]}) = (u,\, u \oplus d)) \,.
\end{aligned}
$$

For any couple $(a,\, b) \in \mathbb{F}_p^2$ and for any $\boldsymbol{r},\, \boldsymbol{r}' \in \mathbb{F}_p^n$, $\boldsymbol{r} \neq \boldsymbol{r}'$, without loss of generality assume $\boldsymbol{r} \neq 0$, there are at most p^{n-1} values of $\boldsymbol{k}$ such that $(\mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r}),\, \mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r}')) = (a,\, b)$. Details on this are given in Appendix A, in particular this justifies the supplementary key addition.

Indeed, it is a system of 2 linear inequivalent equations in $n+1$ independent variables, hence it has $n-1$ degrees of freedom. Thus:

$$\begin{aligned}
\Pr(h_K(\boldsymbol{r})_{[2]} \oplus h_K(\boldsymbol{r}')_{[2]} = d) &\leq \sum_{u \in \mathbb{F}_2^\ell} \sum_{(a,b) \in \mathbb{F}_p^2} \frac{p^{n-1}}{|\Omega|} \Pr((a_{[2]},\, b_{[2]}) = (u,\, u \oplus d)), \\
&\leq \sum_{u \in \mathbb{F}_2^\ell} \sum_{(a,b) \in \mathbb{F}_p^2} \frac{p^{n-1}}{p^{n+1}-p} \Pr(a_{[2]} = u) \Pr(b_{[2]} = u \oplus d), \\
&\leq \sum_{u \in \mathbb{F}_2^\ell} \frac{p^{n+1}}{p^{n+1}-p} \left(\frac{\beta}{p}\right)^2, \\
&= \frac{p^{n+1}}{p^{n+1}-p} \frac{2^\ell \beta^2}{p^2}, \\
&= \frac{1}{p} \frac{2^\ell}{p} \beta^2 \left(1 + \frac{1}{p^n - 1}\right), \\
&= \frac{1}{p} \alpha \beta^2 \,.
\end{aligned}$$

The second line uses that for events A and B, $\Pr(A \cap B) = \Pr(A \mid B) \Pr(B) \leq \Pr(A) \Pr(B)$. With $2^\ell > p$ and $(.)_{[2]}$ the binary decomposition, we have $\beta = 1$, which yields $\Pr(\mathsf{h}_K(\boldsymbol{r})_{[2]} \oplus \mathsf{h}_K(\boldsymbol{r}')_{[2]} = d) \leq \frac{\alpha}{p}$, hence $(\mathsf{h}'_{\boldsymbol{k}})_{\boldsymbol{k}}$ is $\frac{\alpha}{p}$-AXU. □

Note that this proposition works for any ℓ and $x \mapsto x_{[2]}$, including for $2^\ell < p$ and $x \mapsto x_{[2]}$ the reduction modulo 2^ℓ that has $\beta = \lceil \frac{p}{2^\ell} \rceil$, which generalizes the result from Carter and Wegman. Note also that $\alpha \simeq 1$ for p and n not too small and 2^ℓ close to p. From Proposition 5, we have that $(\mathsf{h}'_{\boldsymbol{k}})_{\boldsymbol{k}}$ is $\frac{\alpha}{p}$-almost XOR universal. Using a refinement of the result on the concatenation of universal functions [Sti91], we get the corollary:

Corollary 1 (($\boldsymbol{K} \boxdot \boldsymbol{r})_{[2]}$ is AXU)**.** *Let $\boldsymbol{K}$ be a random matrix of $((\mathbb{F}_p^n \setminus \{0\}) \times \mathbb{F}_p)^m$. Define $\mathsf{h}_{\boldsymbol{K}}(\boldsymbol{r}) = \boldsymbol{K}_{1\ldots n} \cdot \boldsymbol{r} + \boldsymbol{K}_{n+1}$, for $\boldsymbol{r} \in \mathbb{F}_p^n$. The family $(\mathsf{h}'_{\boldsymbol{K}})_{\boldsymbol{K}}$ defined for all $\boldsymbol{K}$ as $\mathsf{h}'_{\boldsymbol{K}}(\boldsymbol{r}) = (((\mathsf{h}_{\boldsymbol{K}}(\boldsymbol{r}))_1)_{[2]}, \ldots, ((\mathsf{h}_{\boldsymbol{K}}(\boldsymbol{r}))_m)_{[2]})$ from $\mathbb{F}_p^n$ into $(\mathbb{F}_2^\ell)^m$ is $\left(\frac{\alpha}{p}\right)^m$-almost XOR universal.*

Proof. Denote $e_K : \mathbb{F}_p^n \to \mathbb{F}_p$ the parallel computation in h', i.e., $e_K(\boldsymbol{r}) = (\sum_{i=1}^n K_i \boldsymbol{r}_i + K_{n+1} \bmod p)_{[2]}$ for K a random variable in $\mathbb{F}_p^{n+1}$. h' is made of m parallel versions of $e_{K^{(i)}}$ with independent keys $K^{(i)}$, but identical input value $\boldsymbol{r}$. Thus for any $\boldsymbol{r} \neq \boldsymbol{r}'$, $e_{K^{(i)}}(\boldsymbol{r}) \neq e_{K^{(i)}}(\boldsymbol{r}')$ for *all* $1 \leq i \leq m$. Thus for any $\boldsymbol{r} \neq \boldsymbol{r}'$,

$$\begin{aligned}
&\Pr((\mathsf{h}'_{K^{(1)}}(\boldsymbol{r}))_1 \oplus (\mathsf{h}'_{K^{(1)}}(\boldsymbol{r}'))_1, \ldots, (\mathsf{h}'_{K^{(m)}}(\boldsymbol{r}))_m \oplus (\mathsf{h}'_{K^{(m)}}(\boldsymbol{r}'))_m = (d^{(1)}, \ldots, d^{(m)})) \\
&\quad \leq \prod_{1 \leq i \leq m} \Pr(e_{K^{(i)}}(\boldsymbol{r}^{(i)}) \oplus e_{K^{(i)}}(\boldsymbol{r}'^{(i)}) = d^{(i)}) \leq \left(\frac{\alpha}{p}\right)^m ,
\end{aligned}$$

as we can apply Proposition 5 on the m independent $e_{K^{(i)}}$ which all have inputs $\boldsymbol{r} \neq \boldsymbol{r}'$. □

From this result, we conclude that $(\mathsf{h}'_{\boldsymbol{K}})_{\boldsymbol{K}}$ is $\left(\frac{\alpha}{p}\right)^m$-almost XOR universal. Therefore $\mathsf{MEDP}((\mathsf{h}'_{\boldsymbol{K}})_{\boldsymbol{K}}) \leq \left(\frac{\alpha}{p}\right)^m$. For p not too small (e.g., the instance of Section 4.3), the factor α can be neglected, as discussed for Proposition 5, and we get $\mathsf{MEDP}((\mathsf{h}'_{\boldsymbol{K}})_{\boldsymbol{K}}) \leq \left(\frac{1}{p}\right)^m$.

Discussion. Security against differential attacks is a standard requirement for cryptographic primitives [BS90]. The MEDP of our construction is close to optimal, which can can be interpreted as follows: set a differential (a, b) through the target function, then a small MEDP implies that for a random choice of key, this differential will have a small probability. This does not guarantee that for any given choice of key, no differential has a high probability. It rather guarantees that the differential (a, b) that maximizes the MEDP will only maximizes the differential probability of a few keys, and that for a given key, few differentials have a high probability. (We checked experimentally that these guarantees were verified for small instances of our construction). Our analysis therefore suggests that for any choice of key, finding a high probability differential is hard.

We note that the supplementary key addition of our construction is required for our proof, which leverages the one of Carter and Wegman (see Appendix A). Without it this result does not hold, and we tested experimentally that the MEDP can be significantly worse in that case, depending on choices of p and n (and can even be equal to one for small n values). It is an interesting open question to find out whether a slightly worse (yet, still sufficient) MEDP could be proven without this supplementary key addition for large enough p and n values. We note also that since our re-keying function works with random inputs, finding a good difference anyway requires birthday complexity (so overall, we do not expect differential attacks to be significant threats against our construction).

We finally mention that truncated differential attacks [Knu94] are of particular interest for our re-keying scheme, as it outputs m words of $\mathbb{F}_p$ by independent parallel computations. It is therefore natural to consider the probability of obtaining a certain difference on one output word, regardless of the rest of the output. From Proposition 5, we have that the MEDP of such a truncated differential is less than $\frac{1}{p}$, which is excellent for an output in $\mathbb{F}_p$, thus discarding the possibility of predicting the output difference.

4.2.2 Linear, algebraic and other cryptanalyses

Security against linear cryptanalysis is another standard requirement for cryptographic primitives [Mat93]. It consists in finding affine relations between the input and output bits of a primitive, that hold with high probability. In turn, it can lead to distinguishers, key-recovery attacks, and also to being able of predicting output bits without the key. Resistance against linear attacks is classically measured with the MELP. However, we are not aware of results bounding the MELP of a universal hash function that would lead to theoretical results similar to the MEDP ones. Hence, we only argue about resistance against linear attacks experimentally. We analyzed reduced instances with $m = 1$ and small primes (i.e., p up to 251) and values of n (i.e., n up to 4) and observed that the MELP follows a similar trend as the MEDP, and is always less than $\frac{1}{p}$ for each output word. We conjecture that for any key, finding a linear relation with high probability is hard.

We also evaluated the algebraic degree of the same small instances as used to heuristically assess the MEDP of our re-keying function and observed that it was always maximum (equal to ℓ), which guarantees that no algebraic attack can be easily performed. In particular, this is different from the case of the scalar product (with or without the supplementary key addition) over $\mathbb{F}_2^\ell$. In this binary field case, the MEDP is still optimal (in particular Proposition 5 holds) but the scalar product is linear. Therefore, higher-order differential attacks [Lai94, Knu94] can be performed exploiting the low degree of the function (which is 1). Having a high degree, our construction over $\mathbb{F}_p$ avoids such issues, and should resist to other attacks exploiting a low-degree function, such as cube attacks [DS09].

Finally, some other techniques seem suitable to analysing our construction. For instance, Divide-and-Conquer techniques seem relevant on the m independent outputs of $\mathsf{h}'_{\boldsymbol{K}}$, and Guess-and-Determine techniques [HR00] could be interesting as the master key $\boldsymbol{K}$ is never modified. Investigating those and more advanced techniques is left for future work.

4.3 Exemplary instance (and variants)

Relying on the previous analyses, we propose to consider an instance of crypto-physical dark matter based on a Mersenne prime $p = 2^{31} - 1$. We aim for the generation of a 128-bit key in the parallel case and therefore consider $m = 4$ (the generated key will have entropy $\approx 4 \times 31$, which is sufficient for the intended applications). As for the size of the secret matrix n, we set it based on the minimum condition $(n + 1) \log p + 3 \log n \geq \lambda$, with $\lambda \approx 128$, and consider $n = 4$. More conservative solutions could be considered (especially for serial implementations, see the discussion in Section 6), but we assume these values are a good starting point in order to stimulate external cryptanalysis. We expect that this instance provides a concrete security such that the first attack path of Figure 3 is more challenging than targeting directly the long-term key shares via a side-channel attack like in Section 5.3, up to significantly higher number of shares than the proposal of Medwed et al. [MSGR10]. Variants that would be worth being investigated in the future include:

- Using a Toeplitz matrix for $\boldsymbol{K}$. This would reduce the key size at the cost of introducing more redundancy in the computations, which could be exploited via mathematical cryptanalysis (extending the results in this paper) or side-channel cryptanalysis (e.g., enabling so-called horizontal attacks [BCPZ16]).

- Removing the additional key addition (i.e., considering a nm-word key rather than a $(n + 1)m$-word key). As mentioned in Section 4.2.1, a first step in this direction would be to evaluate how much the MEDP can be preserved in this case.

5 Implementation results

The previous section showed that a re-keying based on crypto-physical dark matter is significantly more secure than the initial proposal of Medwed et al. [MSGR10], under realistic leakage assumptions. We now discuss the other part of our contribution. Namely, we show that the proposed instance is also significantly more efficient than the wPRF proposal of Dziembowski et al. [DFH$^+$16]. In order to make the two solutions somewhat comparable (given that they are providing security in quite different models), we consider a hashed version of the LWR-based wPRF (that gives a PRF in the random oracle model [BR17]), implemented on a a modern FPGA in [BSS20], which we compare with the full TBC-based re-keying scheme of Figure 1(b), with an AES-based TBC following [LRW11]. We then propose a masked implementation of our new solution relying on a similar hardware architecture as [BSS20] and exhibit the improved performances it achieves. We finally analyze the side-channel security of this masked crypto-physical dark matter and discuss how to select the number of shares in order to reach a given security target.

5.1 Hardware architecture

Our architecture to perform crypto-physical dark matter computations is illustrated in Figure 4. As the architecture in [BSS20], it is designed to leverage the key-homomorphism of the re-keying function by processing the shares serially. It is divided in two main blocks. The first one is composed of the different memories that hold the shares of K, the value of r and the randomness required to refresh the shares. The latter is embedded into the memory blocks and is added to each word of K after it has been read from memory. The second block is the computation core. It includes the logic to perform the different dot product operations (organized as a pipeline for efficiency) and an accumulator that recombines the intermediate results. For security and performance (in particular latency) reasons, we perform 5 multiplications in parallel, leading to an internal bus of 160 bits.

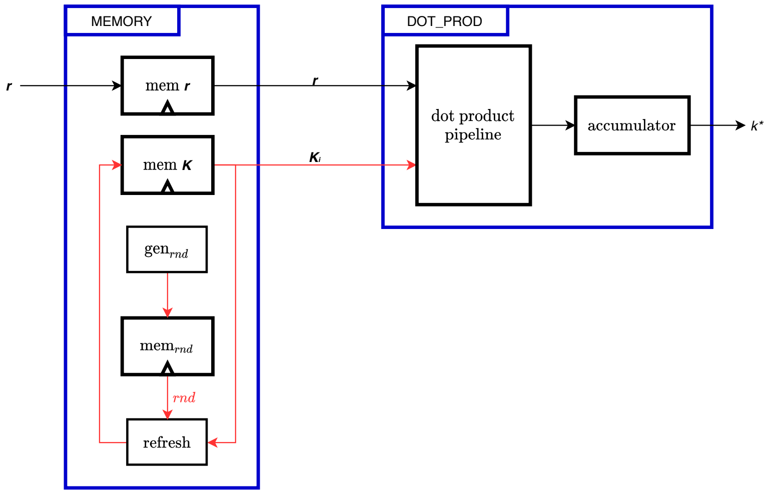


Figure 4: Overview of the crypto-physical dark matter hadrware architecture.

The detailed architecture of the memory block is shown in Figure 5. The r memory consists of a 128-bit long register that is fed with the appropriate value at the beginning of an execution. Both the key and the randomness memories are composed of d (i.e., the amount of shares used) independent memories of $d * m * (n + 1) * 32 = d * 640$ bits. In practice, these are mapped to BRAM blocks which are dedicated memory resources embedded in the FPGA. While processing a specific share, all the memories are read and the appropriate output is selected. The selection mechanism is straightforwardly done using a multiplexer for the randomness memory. An additional register barrier is added in front of the selection multiplexer in order to avoid physical defaults such as glitches that may lead to reductions of the security order [MPG05]. Additionally, the register barriers corresponding to memories that are not expected to be read are reset. The values that are read in memory are then directly forwarded to the computation pipeline.

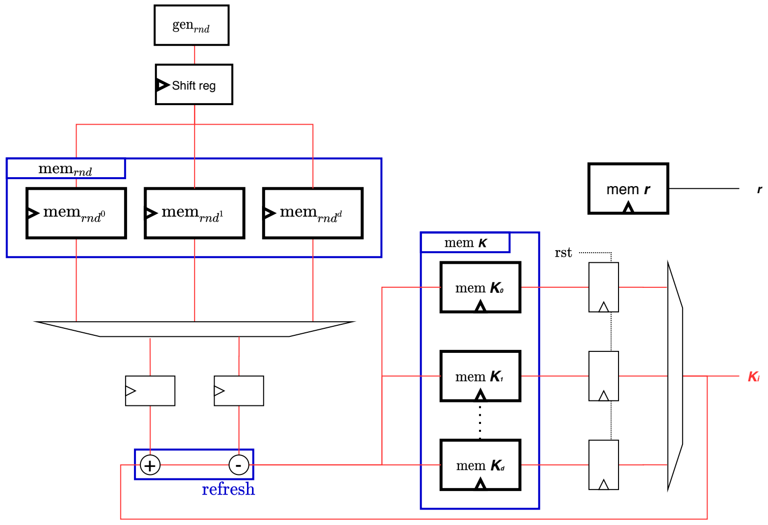


Figure 5: Architecture of the memory block.

A dedicated mechanism is used in order to refresh the key values that are forwarded to the computation core. Each share is directly re-randomized after being used, with a (linear) refresh mechanism that is similar to the one in [BSS20]. Namely, we first generate uniform 31-bit values thanks to four 128-bit LFSRs and output a uniform value over $\mathbb{F}_p$ thanks to a simple rejection sampling (i.e., we output a single 31-bit value that is different from 2^{31}, and use four LFSRs so that they jointly fail with low probability). We then refresh the key words and write them back in the appropriate memory.

The data output by the memory block directly feeds the computation core by entering the modular multiplication layer. As depicted in Figure 6, the latter in composed of 5 independent modular multiplications. We rely on the DSP resources embedded in the FPGA to do so, and more precisely on the multiplier units that they contain.

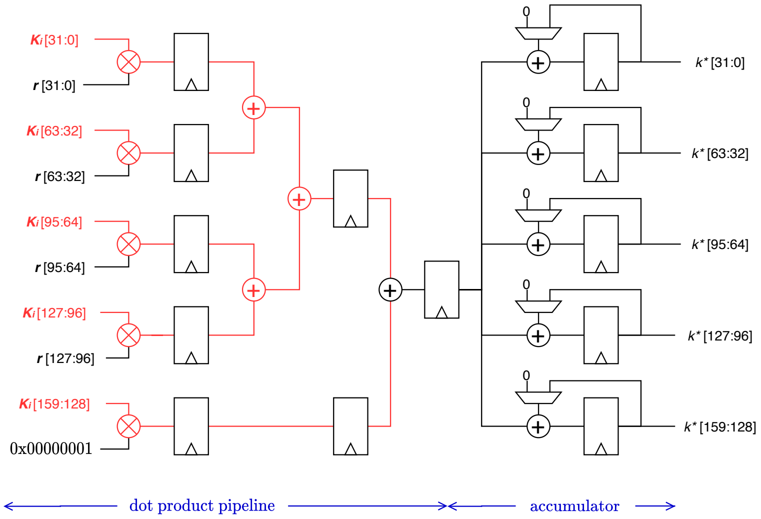


Figure 6: Architecture of the dot product pipeline.

We use instances of the solution presented in [KSHS17] in order to perform the modular multiplications. This solution mixes an efficient utilization of the DSP resources combined with an adder tree specifically designed to limit the logic depth. Additionally, it allows easily incorporating the modular reduction with limited overheads.

Following the modular multiplication, the modular adder tree is split in a two-level pipeline to improve the maximal clock frequency that can be reached. It turns out the register between the second and the third addition layers has a significant impact on the clock frequency, improving its maximal value from 50MHz to 80Mhz at a low logic cost (see Table 1). The difference with the solution in [BSS20] which reaches 90MHz without pipeline is due to the additional logic required to perform the modular reductions.

Eventually, an accumulator ends the pipeline and is composed of 5 independent adders with feedback. As when a refreshed key share is written back in the memory, the data coming from the dot product pipeline is fed to all the adders and only a specific one is activated depending on the line processed. Following this configuration, the final session key value is obtained once all the lines of all the shares have been processed.

5.2 Performance evaluation

We now analyze different performance metrics for our new LWPR-based proposal and compare them with the LWR-based solution of Dziembowski et al. [DFH+16].

We start with the size of the key storage which is among the easiest to quantify. It is represented in Figure 7 and simply illustrates the difference between the (128×32)-bit key that the LWR-based solution requires, to be compared with our (20×32)-bit key.

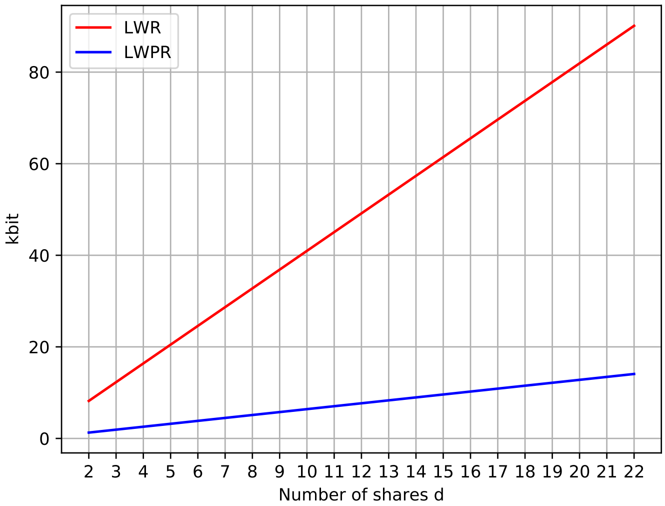


Figure 7: Key storage in function of the number of shares.

We follow with the randomness requirements of the two proposals, which are illustrated in Figure 8. They are proportional to the key size, but the gap between the LWR-based solution and ours is amplified due to the fact that the output of each inner product in [DFH+16] has to be rounded to 10 bits for security reasons, and $\lceil \log_2(d) \rceil$ bits are additionally lost due to the error correcting code that is needed in order to deal with the carry propagations that make the LWR-based wPRF almost key-homomorphic only.

The implementation cost of the two proposals on a Xilinx Kintex 7 is summarized in Table 1. As previously mentioned, our architecture is a parallel one performing $P = 5$ modular multiplications concurrently. The one of [BSS20] uses levels of parallelism $P = 1, 2, 4$ and 8. We report the $P = 4$ and $P = 8$ cases which are the most comparable (since the P parameter also sets the number of DSP blocks used by the FPGA, that is worth $4 \times P$). The figures exclude the key storage that is implemented in the FPGA BRAMs blocks. The LWR-based solution additionally includes an implementation of Keccak to hash the input. The LWPR-based solution additionally includes the implementation of an AES-based TBC. We conclude from these figures that both solutions lead to reasonable costs and their area requirements should not be a problem for practical deployment. We additionally note that in the LWPR case, the cost of the AES-based TBC implementation (which leverages a 32-bit architecture) amounts for 650 registers and 629 LUTs.

Eventually, the most relevant metric to highlight the performance gains of the crypto-physical dark matter is the latency given in Figure 9 (excluding the generation of the masking randomness but including the input hash function for the LWR-based solution, and the AES-based TBC for the LWPR-based one). It underlines that even when the

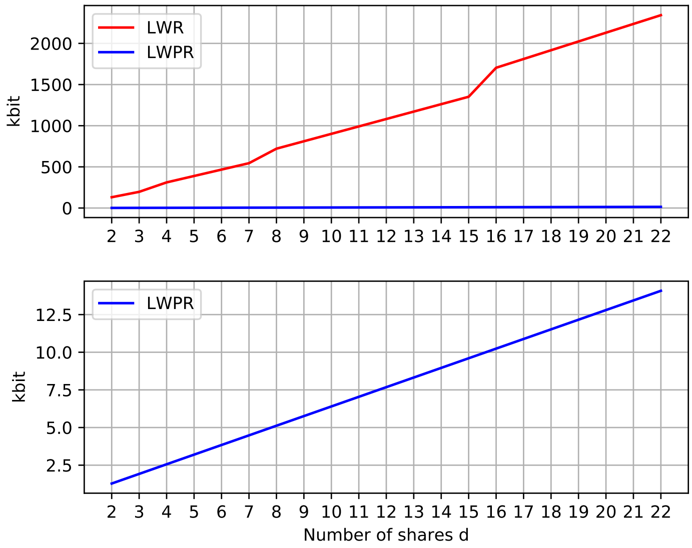


Figure 8: Randomness requirements in function of the number of shares.

Table 1: Implementation cost of the re-keying functions.

	Regs			LUTs		
d	2	4	8	2	4	8
LWR (P=4)	3835	4369	5427	6678	6860	7012
LWR (P=8)	4601	5647	7757	7066	7209	7480
LWPR (no pipe)	1780	2106	2752	3902	4272	8727
LWPR (pipe)	1938	2266	2912	3944	4182	8686

LWR-based solution is implemented with a large level of parallelism (e.g., $P = 8$), the LWPR-based solution is orders of magnitude faster. Furthermore, the lower part of the figure shows that for 20 shares, a full re-keyed TBC can be performed in approximately 160 cycles (to which one must add the cost to generate 12,500 random bits, as per Figure 8, which means a quite reasonable 80 bits per cycle if to be generated on-the-fly).

5.3 Physical security

The previous performance evaluations show that crypto-physical dark matter can be a very competitive option for higher-order masked implementations. For completeness, we next provide the results of a practical security evaluation for such masked implementations, using the same worst-case approach as in [BSS20]. We first recall some generic advantages of key-homomorphic primitives for masking, then describe how to evaluate/bound their concrete (e.g., power consumption) leakage, and finally show how to choose the number of shares needed to reach a given security target thanks to this leakage bound.

Advantages of key-homomorphic primitives for masking. In short, and as carefully discussed in [BSS20], key homomorphic primitives enable a significant simplification of both the design of secure masked implementations and their evaluations.

A first interesting feature for this purpose is that they are trivial to analyze from a probing security viewpoint, since they enable the independent manipulation of the shares. Key-homomorhic primitives do not suffer from composability issues and the only refreshing

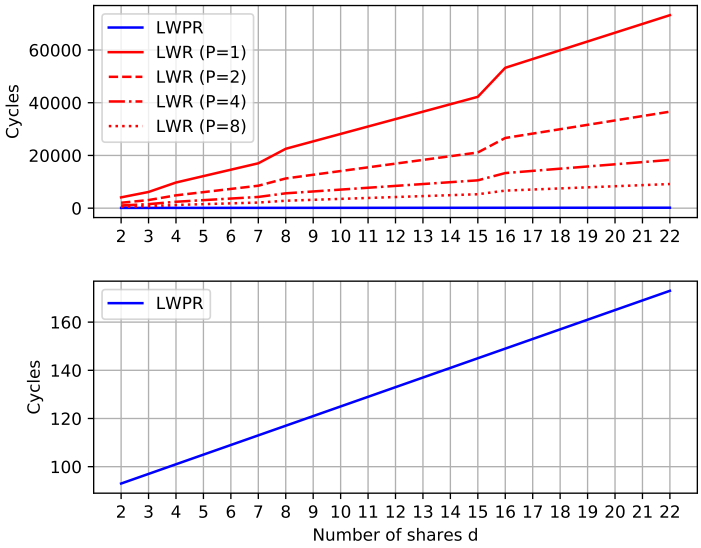


Figure 9: Latency in function of the number of shares.

they need is for the shares of their long-term key, which can be performed using cheap schemes (with linear overheads), as discussed in [BDF+17], Section 8.2.

A second interesting feature is that they mitigate the risks of physical defaults (like glitches or transitions) leading to shares' re-combinations. This benefit again comes from the possibility to manipulate shares independently. In case of shares-serial implementations like the one we chose, they can additionally reduce the risks of couplings [CBG+17].

A third advantage is that each key share is manipulated minimally (i.e., a constant number of times, independent of the security order). This ensures an inherently good resistance against horizontal attacks, formalized by a constant noise rate.

Eventually, a consequence of the previous advantages is that the evaluation of such masked implementations is scalable. For example, increasing the number of shares in a block cipher implementation usually benefits from order-specific optimizations and implies some re-design of the internal components that consequently requires repeating the (time-consuming) side-channel security evaluations for each security order. By contrast, increasing the number of shares of a key-homomorphic architecture boils down to re-using exactly the same component multiple times, avoiding the need to repeat evaluations.

Evaluation of the shares' leakages. The side-channel security of a masked implementation depends on two main assumptions: the shares' leakages must be sufficiently noisy and independent [DFS19]. As discussed and evaluated in [BSS20], the independence is essentially guaranteed by design in a shares-serial key-homomorphic architecture. Since we use the same architecture, we do not detail the detection tests needed to confirm this assumption and rather focus on the level of noise in a prototype implementation.

For this purpose, we first synthetized our design for the Xilinx Spartan 6 FPGA available on the SAKURA-G board.[4] Its clock frequency was set to 6 MHz. The synthesis was performed with the "keep hierarchy" flag avoiding the tool to trim out useful registers. The leakage signal was captured with a Tektronix CT-1 probe. This signal was sampled with a Picoscope 5244d at a rate of 500 MSamples/s, with 12-bit resolution.

An exemplary power traces is given at the top of Figure 10. It corresponds to a 3-share implementation that generates a fresh key in ≈ 12 cycles (i.e., 4 × 3 cycles, where the

[4] http://satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-G.html.

4 factor is the number of 31-bit key words generated to obtain a 124-bit key and the 3 factor is the number of shares). As a standard first step in our evaluations, we estimated the 8-bit side-channel Signal-to-Noise Ratio (SNR) [Man04], which is represented at the bottom of the same figure. We observe that the level of SNR varies between bytes and selected the most leaking byte for our following (worts-case) investigations.[5]

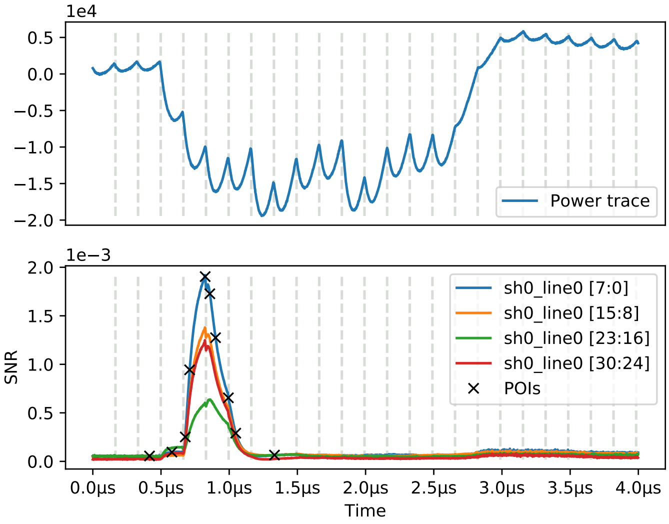


Figure 10: Top: exemplary leakage trace. Bottom: SNR for exemplary target bytes.

As a standard second step in our evaluations, we used the SNR plots to select Points-of-Interest (POIs) in the traces, represented by the crosses in Figure 10. We then estimated the information leakage that can be extracted from the corresponding multivariate distribution under a Gaussian assumption, using the information theoretic metrics and bounds introduced in [BHM+19]. Namely, we estimated the Gaussian Perceived Information (gPI) and the Gaussian hypothetical information (gHI) using the sampling based estimation described in this reference. The convergence of these two metrics for the 10 POIs we selected is illustrated at the top of Figure 11. The gHI gives a bound on the amount of information that can be extracted with such a Gaussian model. We assume that it provides a reasonable approximation of the worst-case security level of our implementation (with the usual cautionary remark that better measurement setups and models can always improve the gPI and its corresponding gHI bound). Since we consider an 8-bit adversary while attacks based on 32-bit guesses can be performed by determined adversaries, we multiplied our leakage estimation by a factor 4 to make our approximations more conservative.

Selecting a number of shares. Eventually, based on the previous estimations of the information leakages and assuming that the independence condition is fulfilled for our implementations, we use the results in [DFS19] in order to select the number of shares needed to reach a target security level. In short, they bound the data complexity N of any side-channel attack against a masked implementation by the inverse of the the mutual information obtained on its shares (which we approximate with the aforementioned gHI) raised to the number of shares d. It leads to the following inequality:

$$N \geq \frac{c}{\mathrm{MI}(\boldsymbol{K};\boldsymbol{L})^d} \approx \frac{c}{\mathrm{gHI}(\boldsymbol{K};\boldsymbol{L})^d},$$

[5] All the SNRs in Figure 10 are for the first share. The following shares give slightly less leakage due to the progressive filling of the pipeline described in Figure 6, which generates algorithmic noise.

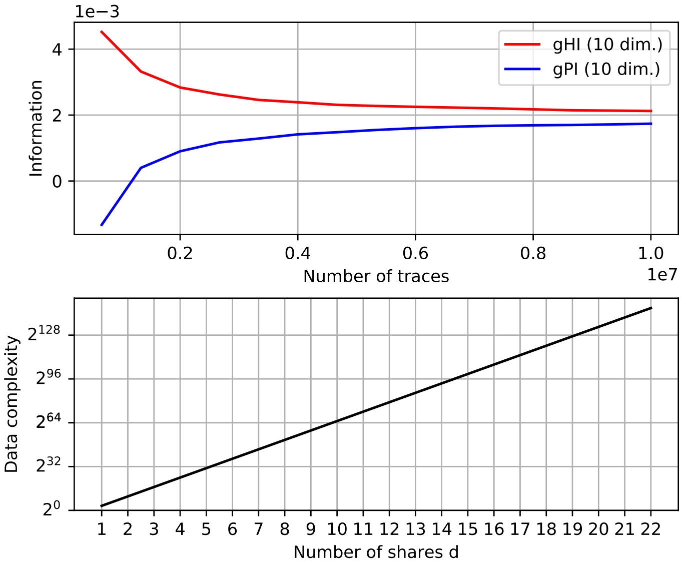


Figure 11: Top: convergence of IT metrics. Bottom: security level.

with c a small constant depending on the size of the key hypothesis and the target success rate [dCGRP19] (e.g., $c = 10$ is a standard value for 8-bit guesses). The resulting data complexities are given at the bottom of Figure 11 in function of the number of shares. They confirm that high security can be obtained against the second attack path of Figure 3.

6 Conclusions & open problems

We introduce crypto-physical dark matter as a provocative solution to improve the security and performances of state-of-the-art re-keying schemes. Its main idea is to combine simple computations in a medium size prime field with a physical leakage function that we assume operating in a sufficiently different field. As feasibility results, we show that such a combination ensures a number of relevant cryptographic properties for the well known Hamming weight leakage function, and that it leads to excellent performances in hardware, leading to a number of stimulating research challenges that we detail next.

From a (both mathematical and implementation) security viewpoint, the preliminary analyzes of this work could be extended towards more advanced attacks, in order to refine the understanding of the complexity to solve the LWPR problem. Another important open question is to generalize our conclusions to broader classes of leakage functions. For example, obtaining results for any "linear" leakage function, as modeled in [SLP05], appears as a natural first step. Eventually, and despite our current security analyzes suggest that some degree of parallelism can make the LWPR problem harder, it would be interesting to study whether crypto-physical dark matter could lead to secure implementations in software (e.g., 32-bit) devices, that are in general difficult to secure against side-channel analysis and will likely require stronger instances and larger number of shares.

Besides, from an application viewpoint, the integration of the proposed re-keying scheme in leakage-resilient modes of operation and/or the efficient protection of decryption algorithms with them would be worth being investigated as well. A natural starting point is the work of Mennink on fresh re-keying applied to authenticated encryption [Men20].

Acknowledgments. Pierrick Méaux is funded by a F.R.S. Incoming Post-Doc Fellowship.

François-Xavier Standaert Standaert is a Senior Research Associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the ERC project 725725 (acronym SWORD) and the Win2Wal project PIRATE.

References

[AKPW13] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. Learning with rounding, revisited - new reduction, properties and applications. In *CRYPTO (1)*, volume 8042 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2013.

[BBD$^+$16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In *ACM Conference on Computer and Communications Security*, pages 116–129. ACM, 2016.

[BCF$^+$15] Sonia Belaïd, Jean-Sébastien Coron, Pierre-Alain Fouque, Benoît Gérard, Jean-Gabriel Kammerer, and Emmanuel Prouff. Improved side-channel analysis of finite-field multiplication. In *CHES*, volume 9293 of *Lecture Notes in Computer Science*, pages 395–415. Springer, 2015.

[BCPZ16] Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In *CHES*, volume 9813 of *Lecture Notes in Computer Science*, pages 23–39. Springer, 2016.

[BDF$^+$17] Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, 2017.

[BFG14] Sonia Belaïd, Pierre-Alain Fouque, and Benoît Gérard. Side-channel analysis of multiplications in GF(2^{128}) - application to AES-GCM. In *ASIACRYPT (2)*, volume 8874 of *Lecture Notes in Computer Science*, pages 306–325. Springer, 2014.

[BGG$^+$14] Josep Balasch, Benedikt Gierlichs, Vincent Grosso, Oscar Reparaz, and François-Xavier Standaert. On the cost of lazy engineering for masked software implementations. In *CARDIS*, volume 8968 of *Lecture Notes in Computer Science*, pages 64–81. Springer, 2014.

[BHM$^+$19] Olivier Bronchain, Julien M. Hendrickx, Clément Massart, Alex Olshevsky, and François-Xavier Standaert. Leakage certification revisited: Bounding model errors in side-channel security evaluations. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 713–737. Springer, 2019.

[BIP$^+$18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: - new simple PRF candidates and their applications. In *TCC (2)*, volume 11240 of *Lecture Notes in Computer Science*, pages 699–729. Springer, 2018.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.

[BR17] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer International Publishing, 2017.

[BS90] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

[BSH$^+$14] Sonia Belaïd, Fabrizio De Santis, Johann Heyszl, Stefan Mangard, Marcel Medwed, Jörn-Marc Schmidt, François-Xavier Standaert, and Stefan Tillich. Towards fresh re-keying with leakage-resilient PRFs: cipher design principles and analysis. *J. Cryptographic Engineering*, 4(3):157–171, 2014.

[BSS20] Olivier Bronchain, Tobias Schneider, and François-Xavier Standaert. Reducing risks through simplicity (high side-channel security for lazy engineers). In *Journal of Cryptographic Engineering*, 2020.

[CBG$^+$17] Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does coupling affect the security of masked implementations? In *COSADE*, volume 10348 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017.

[CCH10a] Claude Carlet, Yves Crama, and Peter L. Hammer. Boolean functions for cryptography and error-correcting codes. In *Boolean Models and Methods*, pages 257–397. Cambridge University Press, 2010.

[CCH10b] Claude Carlet, Yves Crama, and Peter L. Hammer. Vectorial boolean functions for cryptography. In *Boolean Models and Methods*, pages 398–470. Cambridge University Press, 2010.

[CGP$^+$12] Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, Soline Renner, Matthieu Rivain, and Praveen Kumar Vadnala. Conversion of security proofs from one leakage model to another: A new issue. In *COSADE*, volume 7275 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2012.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 1999.

[Cou02] Nicolas Courtois. Higher order correlation attacks, XL algorithm and cryptanalysis of toyocrypt. In Pil Joong Lee and Chae Hoon Lim, editors, *Information Security and Cryptology - ICISC 2002, 5th International Conference Seoul, Korea, November 28-29, 2002, Revised Papers*, volume 2587 of *Lecture Notes in Computer Science*, pages 182–199. Springer, 2002.

[CPRR13] Jean-Sébastien Coron, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche. Higher-order side channel security and mask refreshing. In *FSE*, volume 8424 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 2013.

[CW79] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *J. Comput. Syst. Sci.*, 18(2):143–154, 1979.

[dCGRP19] Eloi de Chérisey, Sylvain Guilley, Olivier Rioul, and Pablo Piantanida. Best information is most successful mutual information and success rate in side-channel analysis. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2019(2):49–79, 2019.

[DEM$^+$17] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, Florian Mendel, and Thomas Unterluggauer. ISAP - towards side-channel secure authenticated encryption. *IACR Trans. Symmetric Cryptol.*, 2017(1):80–105, 2017.

[DEMM14] Christoph Dobraunig, Maria Eichlseder, Stefan Mangard, and Florian Mendel. On the security of fresh re-keying to counteract side-channel and fault attacks. In *CARDIS*, volume 8968 of *Lecture Notes in Computer Science*, pages 233–244. Springer, 2014.

[DFH$^+$16] Stefan Dziembowski, Sebastian Faust, Gottfried Herold, Anthony Journault, Daniel Masny, and François-Xavier Standaert. Towards sound fresh re-keying with hard (physical) learning problems. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 272–301. Springer, 2016.

[DFS19] Alexandre Duc, Sebastian Faust, and François-Xavier Standaert. Making masking security proofs concrete (or how to evaluate the security of any leaking device), extended version. *J. Cryptology*, 32(4):1263–1297, 2019.

[DKM$^+$15] Christoph Dobraunig, François Koeune, Stefan Mangard, Florian Mendel, and François-Xavier Standaert. Towards fresh and hybrid re-keying schemes with beyond birthday security. In *CARDIS*, volume 9514 of *Lecture Notes in Computer Science*, pages 225–241. Springer, 2015.

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Groebner bases. *Journal of Pure and Applied Algebra*, pages 61–88, june 1999.

[GJ19] Qian Guo and Thomas Johansson. A new birthday-type algorithm for attacking the fresh re-keying countermeasure. *Inf. Process. Lett.*, 146:30–34, 2019.

[GMK17] Hannes Groß, Stefan Mangard, and Thomas Korak. An efficient side-channel protected AES implementation with arbitrary protection order. In *CT-RSA*, volume 10159 of *Lecture Notes in Computer Science*, pages 95–112. Springer, 2017.

[GR17] Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 567–597, 2017.

[Hou18] Xiang-dong Hou. *Lectures on Finite Fields*, volume 190. American Mathematical Society, 2018.

[HR00] Philip Hawkes and Gregory G. Rose. Exploiting multiples of the connection polynomial in word-oriented stream ciphers. In *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 303–316. Springer, 2000.

[ISW03] Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

[Knu94] Lars R. Knudsen. Truncated and higher order differentials. In *FSE*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer, 1994.

[KSHS17] Philipp Koppermann, Fabrizio De Santis, Johann Heyszl, and Georg Sigl. Automatic generation of high-performance modular multipliers for arbitrary Mersenne primes on FPGAs. In *HOST*, pages 35–40. IEEE Computer Society, 2017.

[Lai94] Xuejia Lai. *Higher Order Derivatives and Differential Cryptanalysis*, pages 227–233. Springer US, 1994.

[LRW11] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. Tweakable block ciphers. *J. Cryptology*, 24(3):588–613, 2011.

[Man04] Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[Men20] Bart Mennink. Beyond birthday bound secure fresh rekeying: Application to authenticated encryption. *IACR Cryptol. ePrint Arch.*, 2020:1082, 2020.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[MPG05] Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

[MPR+11] 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 *CARDIS*, volume 7079 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2011.

[MSGR10] 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 *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.

[MSJ12] Marcel Medwed, François-Xavier Standaert, and Antoine Joux. Towards super-exponential side-channel security with efficient leakage-resilient PRFs. In *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 193–212. Springer, 2012.

[MSNF16] Marcel Medwed, François-Xavier Standaert, Ventzislav Nikov, and Martin Feldhofer. Unknown-input attacks in the parallel setting: Improving the security of the CHES 2012 leakage-resilient PRF. In *ASIACRYPT (1)*, volume 10031 of *Lecture Notes in Computer Science*, pages 602–623, 2016.

[NRS08] Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of non-linear functions in the presence of glitches. In *ICISC*, volume 5461 of *Lecture Notes in Computer Science*, pages 218–234. Springer, 2008.

[PM16] Peter Pessl and Stefan Mangard. Enhancing side-channel analysis of binary-field multiplication with bit reliability. In *CT-RSA*, volume 9610 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2016.

[PW04] Ruud Pellikaan and Xin-Wen Wu. List decoding of q-ary Reed-Muller codes. *IEEE Trans. Inf. Theory*, 50(4):679–682, 2004.

[SLP05] Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.

[Sti91] Douglas R. Stinson. Universal hashing and authentication codes. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 74–85. Springer, 1991.

[USS+20] Florian Unterstein, Marc Schink, Thomas Schamberger, Lars Tebelmann, Manuel Ilg, and Johann Heyszl. Retrofitting leakage resilient authenticated encryption to microcontrollers. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(4):365–388, 2020.

[Vau99] Serge Vaudenay. On the security of CS-Cipher. In *FSE*, volume 1636 of *Lecture Notes in Computer Science*, pages 260–274. Springer, 1999.

A Supplementary key addition

For any couple $(a, b) \in \mathbb{F}_p^2$ and for any $\boldsymbol{r}, \boldsymbol{r}' \in \mathbb{F}_p^n$, $\boldsymbol{r} \neq \boldsymbol{r}'$, wlog assuming $\boldsymbol{r} \neq 0$, there are at most p^{n-1} values of $\boldsymbol{k}$ such that $(\mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r}), \mathsf{h}_{\boldsymbol{k}}(\boldsymbol{r}')) = (a, b)$. We have equations in p^{n+1} variables $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_{n+1}$, and a system of two equations:

$$\begin{cases} <\boldsymbol{k}_{1\ldots n}, \boldsymbol{r}> +\boldsymbol{k}_{n+1} &= a, \\ <\boldsymbol{k}_{1\ldots n}, \boldsymbol{r}'> +\boldsymbol{k}_{n+1} &= b \end{cases}$$

$$\Leftrightarrow \begin{cases} <\boldsymbol{k}_{1\ldots n}, \boldsymbol{r}> +\boldsymbol{k}_{n+1} &= a, \\ <\boldsymbol{k}_{1\ldots n}, \boldsymbol{r}' - \boldsymbol{r}> &= b - a \end{cases}$$

As long as the two equations are not colinear and non-trivial, this system is of rank 2, hence it has at most $\frac{|\Omega|}{p^2} \leq p^{n-1}$ solutions. The supplementary key addition makes that these equations cannot be colinear, as only the first one depends on $\boldsymbol{k}_{n+1}$.

Besides, these equations are also non-trivial (i.e., they depend on the key), since the first one depends on $\boldsymbol{k}_{n+1}$ and in the second one, $\boldsymbol{r}' - \boldsymbol{r} \neq 0$.

By contrast, without the supplementary key addition, for choices of $\boldsymbol{r}, \boldsymbol{r}'$ such that $\boldsymbol{r}_i = c\boldsymbol{r}'_i$ for all i, there exist $p-1$ couples $(a, b) = (a, ca)$ such that the two equations are colinear (unless $c = 0$ in which case only the couple $(a, b) = (0, 0)$ is a solution), hence the system becomes of rank 1 and has at most $\frac{p}{|\Omega|}$ solutions for all such couples (a, b).