Leakage Squeezing of Order Two

Claude CARLET[1], Jean-Luc DANGER[2,3],
Sylvain GUILLEY[2,3] and Houssem MAGHREBI[2].

[1] LAGA, UMR 7539, CNRS, Department of Mathematics,
University of Paris XIII and University of Paris VIII,
2 rue de la liberté, 93 526 Saint-Denis Cedex, France.

[2] TELECOM-ParisTech, Crypto Group,
37/39 rue Dareau, 75 634 Paris Cedex 13, France.

[3] Secure-IC S.A.S., 80 avenue des Buttes de Coësmes,
35 700 Rennes, France.

claude.carlet@univ-paris8.fr,
{jean-luc.danger,sylvain.guilley,houssem.maghrebi}@telecom-paristech.fr

Abstract. In masking schemes, *leakage squeezing* is the study of the optimal shares' representation, that maximizes the resistance order against high-order side-channel attacks. Squeezing the leakage of first-order Boolean masking has been problematized and solved previously in [10]. The solution consists in finding a bijection F that modifies the mask, in such a way that its graph, seen as a code, be of greatest dual distance. This paper studies second-order leakage squeezing, *i.e.* leakage squeezing with two independent random masks. It is proved that, compared to first-order leakage squeezing, second-order leakage squeezing at least increments (by one unit) the resistance against high-order attacks, such as high-order correlation power analyses (HO-CPA). Now, better improvements over first-order leakage squeezing are possible by relevant constructions of the squeezing bijections pair. We provide with linear bijections that improve by strictly more than one (instead of one) the resistance order. Specifically, when the masking is applied on bytes (which suits AES), resistance against 1st-order (resp. 2nd-order) attacks is possible with one (resp. two) masks. Optimal leakage squeezing with one mask resists HO-CPA of orders up to 5. In this paper, with two masks, we provide resistance against HO-CPA not only of order $5 + 1 = 6$, but also of order 7.

Keywords: High-order side-channel attacks, leakage squeezing, Boolean logic, rate 1/3 linear codes with 3 disjoint information sets, AES.

1 Introduction

Masking is an implementation-level strategy to thwart side-channel attacks. A dth-order masking scheme consists in replacing the manipulation of one sensitive variable X by the manipulation of a vector of $d + 1$ variables $S_0, \cdots, S_d$ called shares, in such a way that:

– X can be deterministically reconstructed from all the shares, while
– no information on X can be retrieved knowing strictly less than $d+1$ shares.

In this case, sometimes referred to as *perfect masking*, it has been shown that:

– arbitrary computations can be carried out (see for instance [23]), and that
– the leaked information is nonzero, but decreases exponentially as $\mathcal{O}\left(\sigma^{-2\times d}\right)$, where σ^2 is the variance of the noise that characterizes the measurement process [8].

Besides, it has been often reported that the cost overhead of masking, in terms of program executable file size or running time for software applications and in terms of implementation area for hardware applications, is too high for its adoption in real-world products. Therefore, the optimization of masking is of great practical importance.

The typical behavior of computing devices is to leak a non-injective and noisy function of the shares. It is usually modeled as a deterministic function of the shares plus an additive white Gaussian noise (AWGN). This model is justified by the fact that an attacker can only measure an aggregated function of each computing element's leakage, such as the total current drawn by the circuit. This means that the measurement indeed consists in the sum of the individual leakages of each processed bit, that can be partitioned into:

– the sum of the individual leakages of the bits of the sensitive variable X (which is obviously non-injective, as it projects words of identical Hamming weight onto the same image), and
– the sum of the individual leakages of each non-sensitive variable bits (that obeys a multinomial distribution, well approximated by a normal law).

Depending on the execution platform, the leakage of one bit can be modelled according to:

– its activity (the leakage is observed when the bit changes values), or
– its value (the leakage differs according to the bit's state).

Without loss of generality, we assume the first kind of leakage, which corresponds to the behavior of CMOS logic. The second kind of leakage is a particular case where the previous value is constant and null. Additionally, it can be assumed that every bit of a sensitive variable leaks an identical amount, irrespective of its neighbors. These assumptions lead to the so-called Hamming distance leakage model, *i.e.* a model in which the attacker records the noisy version of the sum of bitflips occurring in X.

This model might not comply exactly with the actual real-world leakage. One research direction is to study the impact of imperfections in the model (because of chip's design variability), that can be quantified for instance with the "perceived information" [22] metric. Another research direction is to do the most of "off-the-shelf" imperfect hardware. For instance, in the case when the countermeasure designer can influence the chip's manufacturing, he can ask that

the indistinguishability of the bits and their non-interference explicitly figure in the product specifications. Technically, these requirements can be met; as a matter of fact, the gates that hold the bits of X:

- can be different instances of the same register flip-flop, constrained to have an identical fanout, and
- can be placed far away one from each other, with their output routing wires adequately shielded from nearby aggressors, so as to reduce their cross-talk.

Experimental feedback (from *in silico* measurements) indicates that those constraints are realistic [27]; for instance, in dual-rail logics, such constraints are enforced [26], with varied efficiency in a "static setup" (*i.e.* the only entropy comes from the data). However, in a "dynamic setup", such as masking, these constraints can definitely improve the trustworthiness on the accuracy of the leakage model.

Glitching is another flaw that limits the efficiency of the masking countermeasures; it is a "logical" coupling (as opposed to the "technological" nature of the cross-talk) that produces a higher-order leakage, not captured in the model. For example, it is reported in [15] a glitch that combines the two shares of a first-order countermeasure, thereby unintentionally disclosing one bit of X through the leakage function $\mathscr{L}$. The designer can opt to hide the computations in a synchronous memory table, that evaluates the output at once [25]. In such condition, no glitch is possible, since glitches stem from a race between two signals that converge to the inputs of a gate. However, tables are expensive. Nonetheless, it is possible to break the tables into smaller elements, provided each of them remain glitch-free. This is possible if every computing element receives its inputs simultaneously. Such strategy can be implemented at the gate-level if every gate is clocked and the combinational logic behaves like a very fast pipeline, as explained in [16]. Also, the designer can take advantage of recent works about "threshold implementations" [18] or "multi-party computation" [21], that both aim at securing masked combinational logic against insidious leakage conveyed by glitches. Their principle is to partition the combination functions into non-interfering submodules that compute on d shares or less, which denies all possibility of glitchy recombination that could disclose (all or part of) the sensitive variable X. In the other case when the countermeasure designer must use an already hard-wired circuit, then profiling can be used to characterize to which extent the leakage conditions are satisfied. The stochastic method [24] allows to precisely assess the leakage model. Notably, first-order coefficients should be checked to be as equal one to each other as possible, and second-order coefficients as small as possible [6] with respect to first-order coefficients. Eventually, it is known that in implementations with combinational logics (*e.g.* the sbox of the DPA contest v2 [4]), the Hamming distance 0 signs much less than the others. The reason is that the combinational nets of the sbox are already prepositioned, and thus the next identical computation does not require to recompute them. On tables, this "memory effect", also termed "clockwise collision" [7], is less visible, as all accesses, even identical consecutive ones, draw some current due to the dynamic character of lookups.

In the sequel, we assume that the leakage $\mathscr{L}$ is equal, or close enough, to the assumed model. In this case, the security of the masking countermeasure can be greatly enhanced. Notably, the indiscernibility and the non-interference of the bits can be taken advantage of to reach $(d+1)$th-order security with strictly less than $d+1$ shares. This strategy is called "leakage squeezing" [12]. It can be seen as a constructive combination of *masking* (through the splitting of X into shares) and of *hiding* (through the leakage function $\mathscr{L}$ properties, namely the leakage in Hamming distance). The figure 1, whose layout is inspired from [14, p. 12], illustrates the symbiosis of the *masking* and *hiding* countermeasures tactics in the leakage squeezing. Roughly speaking, masking is a "software" countermeasure, in that it is implemented by the designer (in assembly language or hardware description language), whereas hiding is a "hardware" countermeasure, in that it is a native property of the device.

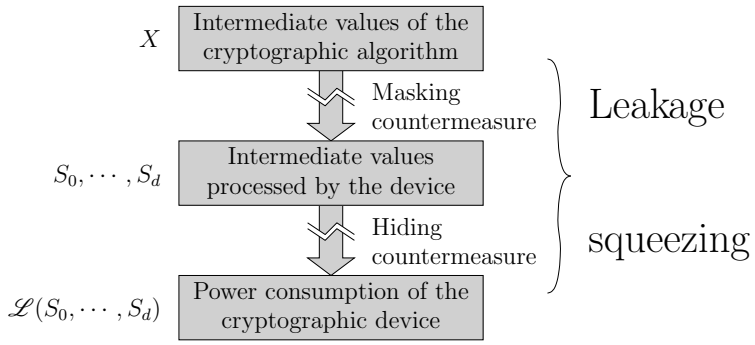

Fig. 1. Principle of the leakage squeezing, that takes on attributes from both the "masking" and the "hiding" strategies.

A masking scheme involves a group $(\mathcal{X}, \perp)$, where $\mathcal{X}$ is the support of the sensitive variable X and $\perp$ an internal composition law. By definition of a group, the zero element 0 is neutral, *i.e.* $\forall X \in \mathcal{X}, X \perp 0 = 0 \perp X = X$, and for all element $X \in \mathcal{X}$, there is an opposite element denoted by $-X \in \mathcal{X}$ that satisfies $X \perp -X = -X \perp X = 0$. Several conventions can be adopted; in the most commonly encountered one, the sensitive variable is obtained as $X = S_0 \perp \cdots \perp S_d$. Under this assumption, $S_1, \cdots, S_d$ are independent uniformly distributed random variables on $\mathcal{X}$, and $S_0 \doteq X \perp \perp_{i=1}^{d} -S_i$. In digital circuits, the set $\mathcal{X}$ is made up of vectors of n bits. For example, $n = 8$ in AES, that manipulates bytes; also, $n = 4$ for DES, since it is usually the output of the sboxes that are targeted. Classical examples of masking are:

- Boolean masking, with $(\mathbb{F}_2^n, \oplus)$, or
- arithmetic masking, with $(\mathbb{Z}_{2^n}, \boxplus)$, where $\boxplus$ represents the modular addition.

In this paper, we will be making use of Boolean masking, as it lessens the degradation of performances in the context of hardware implementations: bits are masked one by one, hence the impact of the masking on the critical path is lowered (in particular, we avoid the carry propagation inherent to the arithmetic masking). Also, in $\mathbb{F}_2^n$, the opposite of a share $-S_i$ is the share itself ($-S_i = S_i$), hence the masking and demasking hardware can be factored.

The rest of this paper is structured as follows. In Sec. 2, we explain briefly the rationale about first-order leakage squeezing. Its extension to second-order leakage squeezing is tackled with in Sec. 3. In this section, we show that this generalization is not trivial. Nonetheless, we manage to characterize the adequate bijections and present some interesting solutions. Eventually, conclusions and perspectives are in Sec. 4. A case study on linear second-order leakage squeezing is given in Appendix A for $n = 8$ and Appendix B for $n = 4$. These two last sections detail some practicalities: the article remains self-contained even without reading them.

2 Reminder on Leakage Squeezing

In this section, we recall the prior art on leakage squeezing at the destination of the reader who is not already acquainted with the notions introduced in [10] by Maghrebi *et al.* The gist of the article is Sec. 3; so this section can be safely skipped by the reader interested mainly in the progress over the state-of-the-art.

2.1 Leakage Squeezing in the Hamming Distance Model

The principle of first-order leakage squeezing is sketched in Fig. 2. The functional computation is carried out on the sensitive variable X, that is mixed with a random uniformly distributed mask (also of n-bit size) denoted by M. The shares are $(S_0, S_1) = (X \oplus M, M)$. As opposed to straightforward first-order masking, the shares are not held as such in registers. Instead, the two registers contain $X \oplus M$ (*i.e.* S_0) and $F(M)$ (*i.e.* $F(S_1)$). The function F must be a bijection, so that the mask value can be recovered from $F(S_0)$. In Fig. 2, the computational logic is concealed in memory tables (to ensure a glitch-free computation). However, any other "more optimized" (tables with $2n$-bit addresses are expensive) logic would also be suitable. The computation is conducted in such a way to respect the invariant:

$$X = \underbrace{S_0}_{\text{Masked data path}} \oplus \underbrace{S_1}_{\text{Mask path}} . \tag{1}$$

The scheme presented in Fig. 2 allows to compute (X', M') from (X, M) in one clock cycle:

- $X' = C(X)$ is a combinational function of X, where $C : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is the expected functionality,

– $M' = R(M)$ is the mask refresh function. Two options are possible: either the mask M' is derived from M deterministically through $R : \mathbb{F}_2^n \to \mathbb{F}_2^n$, or it disregards M and it is drawn fresh from a true random number entropic source.

After one iteration, the invariant condition of Eqn. (1) is still met: $X' = S'_0 \oplus S'_1 = (X' \oplus M') \oplus F^{-1} \circ F(M')$.

In a hardware setup, the shares leak in the Hamming distance model. The leakage is thus equal to $\mathscr{L} = \mathsf{HW}((X \oplus M) \oplus (X' \oplus M')) + \mathsf{HW}(F(M) \oplus F(M'))$, that can be rewritten as $\mathscr{L} = \mathsf{HW}(Z \oplus M'') + \mathsf{HW}(F(M) \oplus F(M \oplus M'')) = \mathsf{HW}(Z \oplus M'', D_{M''}F(M))$. In this expression:

– HW is the Hamming weight function,
– Z is the difference between two consecutive values of the masked data ($Z \doteq X \oplus X'$),
– M'' is the difference between two consecutive values of the mask ($M'' \doteq M \oplus M'$) and
– $D_Y F(X)$ is the Boolean derivative of F in direction Y taken at point X.

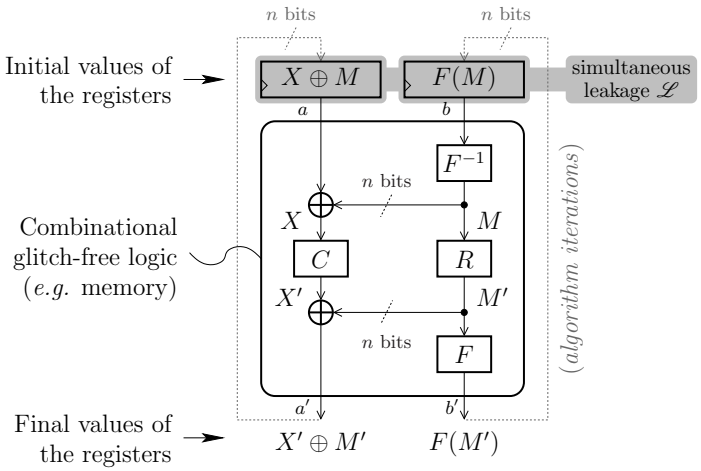

Fig. 2. Setup of the first-order masking countermeasure with bijection F.

In the rest of this section, we recapitulate in one single page the key steps described extensively in [10] to find the first-order optimal leakage squeezing. The paper [10] is thus hereafter only surveyed, to highlight the reasoning. The section 3 will conduct step-by-step an accurate and self-contained analysis of the two-mask case.

It is shown in [10] that this leakage function is unexploitable by a dth-order correlation power analysis if all the terms $\mathbb{E}[\mathsf{HW}(Z \oplus M'')^p \times \mathsf{HW}(D_{M''}F(M))^q \mid$

$Z = z]$, whatever p, q such as $p + q \leq d$ do not depend on z. In this expression, the capital letters represent random variables, and $\mathbb{E}$ is the expectation operator. The condition on F is equivalent to finding a bijection $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ that satisfies:

$$\forall a \in \mathbb{F}_2^{n*}, \quad \widehat{\mathsf{HW}^p}(a) = 0 \quad \text{or} \quad \mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(\cdot)}F(M)}](a) = 0 \ . \tag{2}$$

The term HW^p (resp. HW^q) represents the Hamming weight function raised at the power of $p \in \mathbb{N}$ (resp. $q \in \mathbb{N}$). The "hat" symbol represents the Fourier transform, that turns a function $f : \mathbb{F}_2^n \to \mathbb{Z}$ into $\hat{f} : \mathbb{F}_2^n \to \mathbb{Z}, x \mapsto \sum_{y \in \mathbb{F}_2^n} f(y)(-1)^{y \cdot x}$. Eventually this expression, $\mathbb{E}[\mathsf{HW}^q \circ D_{(\cdot)}F(M)]$ designates the function:

$$\begin{aligned} \mathbb{E}[\mathsf{HW}^q \circ D_{(\cdot)}F(M)] : \mathbb{F}_2^n &\to \mathbb{Z} \\ m'' &\mapsto \mathbb{E}[\mathsf{HW}^q \circ D_{m''}F(M)] = \tfrac{1}{2^n} \textstyle\sum_m \mathsf{HW}^q(D_{m''}F(m)) \ . \end{aligned}$$

The Eqn. (2) can be simplified, as Theorem 1 below is proved in [10, Appendix A.1].

Theorem 1. $\forall a \in \mathbb{F}_2^n, \forall p \in \mathbb{N}, \quad \widehat{\mathsf{HW}^p}(a) = 0 \iff \mathsf{HW}(a) > p \ .$

So the condition for the leakage squeezing to reach order d is simply to have: for all $a \in \mathbb{F}_2^{n*}$ and for all p such that $\mathsf{HW}(a) \leq p$ and for all q such as $q \leq d - p$, $\mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(\cdot)}F(M)}](a) = 0$.

This condition is also equivalent to (refer to forthcoming Lemma 1 at page 11):

$$\forall p, \forall (a, b) \text{ such that } \mathsf{HW}(a) \leq p \text{ and } \mathsf{HW}(b) \leq d - q, \text{ we have } \widehat{(b \cdot F)}(a) = 0 \ .$$

As shown in [10, Sec. 4], this condition can be related to "complementary information set" codes (*also known as* CIS codes [2]). It is equivalent that the indicator of the graph $\{(x, F(x)); x \in \mathbb{F}_2^n\}$ of F is d-th order correlation immune.

2.2 Leakage Squeezing in the Hamming Weight Model

If the device leaks in Hamming weight, then the relations are still valid if we replace the derivative $D_{(\,\cdot\,)}F$ of F by F itself. Such an analysis is conducted in [8]. It is also worthwhile mentioning that if F is linear, the two problems are the same, because $D_m F(x) = F(x \oplus m) \oplus F(x) = F(x \oplus m \oplus x) = F(m)$, irrespective of x. This property is important, as a recent scholar work has shown empirically that on FPGAs, both Hamming distance and Hamming weight leakage models should be envisioned [17].

3 Second-Order Leakage Squeezing

3.1 Goal

In this section, an improvement of the leakage squeezing where two masks are used is studied. More precisely,

– the masked data ($X \oplus M_1 \oplus M_2$, also noted $X \oplus M$, where $M \doteq M_1 \oplus M_2$) is processed as is, *i.e.* through a bijection that is the identity (denoted by Id),
– the first mask (M_1) is processed through bijection F_1 and
– the second mask (M_2) is processed through bijection F_2.

This second-order masking scheme is illustrated in Fig. 3. With respect to the first-order masking scheme (Fig. 2 – described in § 2.1), the processing of the masked sensitive data is unchanged, and only the masks processing differs: each mask can be seeded independently and evolves from a different diversification function (noted R_1 and R_2).

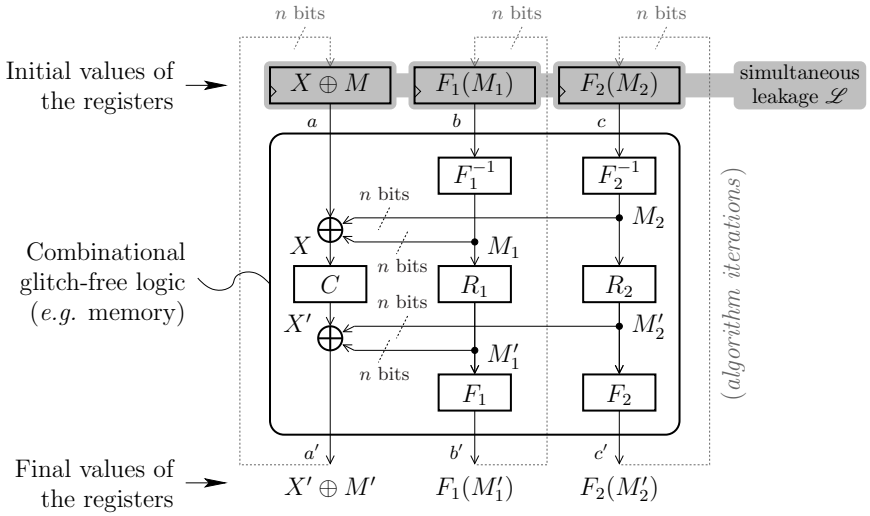

Fig. 3. Setup of the second-order leakage squeezing masking countermeasure with bijections F_1 and F_2.

The leakage function is thus:

$$\mathscr{L} = \mathsf{HW}((X \oplus M_1 \oplus M_2) \oplus (X' \oplus M'_1 \oplus M'_2), F_1(M_1) \oplus F_1(M'_1), F_2(M_2) \oplus F_2(M'_2)) \ .$$

As previously, $Z \doteq X \oplus X'$, and furthermore we denote: $M''_1 \doteq M_1 \oplus M'_1$ and $M''_2 \doteq M_2 \oplus M'_2$. Hence the leakage:

$$\begin{aligned} \mathscr{L} &= \mathsf{HW}(Z \oplus M''_1 \oplus M''_2, F_1(M_1) \oplus F_1(M_1 \oplus M''_1), F_2(M_2) \oplus F_2(M_2 \oplus M''_2)) \\ &= \mathsf{HW}(Z \oplus M''_1 \oplus M''_2, D_{M''_1} F_1(M_1), D_{M''_2} F_2(M_2)) \ . \end{aligned} \tag{3}$$

3.2 Motivation

It could be argued that the security brought by first-order leakage squeezing is already high enough, and resisting at still higher orders is a superfluous refine-

ment. Admittedly, it has seldom been question of high-order attacks of order strictly greater than two in the abundant public literature.

However, searching for greater security can be motivated by "forward security" concerns. Secure elements (*e.g.* smartcards, RFID chips, hardware security modules, *etc.*) contain high-value secrets, and cannot be upgraded. Therefore, one can imagine buying one of these today, and having it attacked with tomorrow's know-how. For instance, with the advance of science, measurements apparati will have a lower noise figure and a greater vertical resolution in the future, thereby reducing the noise in side-channel acquisitions. Now, it is known that the limiting factor for the high-order attacks is the noise variance [29]. Also, it is now well understood how to combine partially successful side-channel attacks with brute force search [3,28]. Therefore, computer-assisted side-channel attacks might greatly enhance what can be done today. Thus, to avoid tomorrow successful attacks of orders greater of one, two, or more orders than what is possible today, precautions must be envisioned today. A parallel can be made with the evolution of:

- the key size of block ciphers,
- the modulus size of asymmetric primitive, or
- the internal state of hash functions.

Those have continuously been increasing over the last years. Besides, the regulation in terms of security compliance standards is always one step ahead the state-of-the-art attacks. Consequently, it is not absurd that side-channel resistance of very high order be demanded soon (*e.g.* with the forthcoming standard ISO/IEC 17 825), hence an incentive for the research in really high-order countermeasures.

Finally, some products supporting second-order countermeasures are already deployed in the field. The second-order leakage squeezing can be mapped in the devices of this installed base at virtually no extra cost, and so the application of this method in real products does not require further architectural development costs. The sole modification is the entry of the masking material in the (F_1, F_2) bijections, and their leaving at the end of the cryptographic application.

3.3 Formalization of Second-Order Leakage Squeezing

The attack fails at order d if $\forall i \leq d, \mathbb{E}\left((\mathscr{L} \mid Z = z)^i\right) = \mathbb{E}\left(\mathscr{L}^i \mid Z = z\right)$ does not depend on z. Indeed, the attacker has thus no bias to relate the leakage at order $i \geq 1$ to the (predictable and key-dependent) sensitive variable Z. Now, the goal of the attacker is to exhibit a bias in $\mathbb{E}\left(\mathscr{L}^d \mid Z = z\right)$ for an exponent d as small as possible, because the noise in $\mathscr{L}^d$ evolves as $\left(\sigma^2\right)^d$ [29], where σ^2 is the variance of the noise (for $d = 1$). So the smaller d, the greater the signal-to-noise ratio, on which depends the number of traces to extract the correct key [13]. Taking into account the formula of $\mathscr{L}$ from Eqn. (3), we have the following

expression for $\mathbb{E}\left(\mathscr{L}^i \mid Z = z\right)$:

$$\begin{aligned}
&\mathbb{E}\left(\left(\mathsf{HW}(Z \oplus M_1'' \oplus M_2'', D_{M_1''}F_1(M_1), D_{M_2''}F_2(M_2))\right)^i \mid Z = z\right)\\
&= \frac{1}{2^{4n}} \sum_{m_1'', m_2''} \sum_{m_1, m_2} \left(\mathsf{HW}(z \oplus m_1'' \oplus m_2'', D_{m_1''}F_1(m_1), D_{m_2''}F_2(m_2))\right)^i\\
&= \frac{1}{2^{4n}} \sum_{\substack{m_1'', m_2''\\ m_1, m_2}} \left(\underbrace{\mathsf{HW}(z \oplus m_1'' \oplus m_2'')}_{\text{Term \#0}} + \underbrace{\mathsf{HW}(D_{m_1''}F_1(m_1))}_{\text{Term \#1}} + \underbrace{\mathsf{HW}(D_{m_2''}F_2(m_2))}_{\text{Term \#2}} \right)^i .
\end{aligned}$$

This equation can be developed, to yield a sum of products of the three terms. Let us denote by p, q and r the degrees of each term, that satisfy $p+q+r=i$. So attacks fail at order d if for all p, q and r such as $p+q+r \leq d$, the function

$$\begin{aligned}
&z \mapsto f(z)\\
&\doteq \sum_{m_1'', m_2''} \sum_{m_1, m_2} \mathsf{HW}^p(z \oplus m_1'' \oplus m_2'') \cdot \mathsf{HW}^q(D_{m_1''}F_1(m_1)) \cdot \mathsf{HW}^r(D_{m_2''}F_2(m_2))\\
&= \sum_{m_1'', m_2''} \mathsf{HW}^p\left(z \oplus m_1'' \oplus m_2''\right) \cdot \sum_{m_1} \mathsf{HW}^q\left(D_{m_1''}F_1(m_1)\right) \cdot \sum_{m_2} \mathsf{HW}^r\left(D_{m_2''}F_2(m_2)\right)\\
&= \sum_{m_1'', m_2''} \mathsf{HW}^p\left(z \oplus m_1'' \oplus m_2''\right) \cdot \mathbb{E}[\mathsf{HW}^q\left(D_{m_1''}F_1(M_1)\right)] \cdot \mathbb{E}[\mathsf{HW}^r\left(D_{m_2''}F_2(M_2)\right)]\\
&= \left\{\mathsf{HW}^p \ \otimes\ \mathbb{E}[\mathsf{HW}^q \circ D_{(\cdot)}F_1(M_1)] \ \otimes\ \mathbb{E}[\mathsf{HW}^r \circ D_{(\cdot)}F_2(M_2)]\right\}(z) \qquad (4)
\end{aligned}$$

is constant. From Eqn. (4), we see that every term to be kept constant is a double convolution product.

Keeping f constant is equivalent to having the Fourier transform $\hat{f}$ of f null everywhere but in zero. The Fourier transform turns a convolution product into a product; therefore,

$$\hat{f} = \widehat{\mathsf{HW}^p} \cdot \mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(\cdot)}F_1}(M_1)] \cdot \mathbb{E}[\widehat{\mathsf{HW}^r \circ D_{(\cdot)}F_2}(M_2)] \ .$$

In summary, to resist at order d, we are attempting to find two bijections F_1 and F_2 such as:

$$\begin{aligned}
\forall a \in \mathbb{F}_2^{n*}, \quad \widehat{\mathsf{HW}^p}(a) = 0 \ \text{ or } \ \mathbb{E}[\widehat{\mathsf{HW}^q \circ D_{(\cdot)}F_1}(M)](a) = 0\\
\text{or } \ \mathbb{E}[\widehat{\mathsf{HW}^r \circ D_{(\cdot)}F_2}(M)](a) = 0 \ , \qquad (5)
\end{aligned}$$

for every triple of integers p, q and r such as $p+q+r \leq d$, d being the targeted protection order.

The Fourier support of a function $\psi : \mathbb{F}_2^n \to \mathbb{Z}$ is the set $\left\{a \in \mathbb{F}_2^n ; \hat{\psi}(a) \neq 0\right\}$. The equation (5) expresses the fact that the Fourier supports of HW^p, $\mathbb{E}[\mathsf{HW}^q \circ D_{(\cdot)}F_1(M)]$ and $\mathbb{E}[\mathsf{HW}^r \circ D_{(\cdot)}F_2(M)]$ intersect only in the singleton $\{0\}$.

3.4 Gaining At Least one Order With Two Masks instead of One

It is a well known property that adding one mask increases the security by one order [29]. We here prove that the same benefit can be expected from the leakage squeezing. We refer to the special case when the second mask is used without transformation (*i.e.* $F_2 = \mathsf{Id}$) as "partial leakage squeezing of order two".

Proposition 1. *Let F_1 be a bijection such that the security is reached at order d with one mask. Then, by introducing a second mask processed through whatever bijection F_2, the security is reached at order at least $d+1$.*

Proof. Let (p, q, r) be any triple of integers such as $p + q + r \leq d + 1$. Then:

- if $r = 0$, $\widehat{\mathsf{HW}^r \circ D_{(\cdot)} F_2} = \widehat{1 \circ D_{(\cdot)} F_2} = \widehat{1} = \delta$ is a Kronecker symbol function, hence null for all $a \neq 0$,
- otherwise, $r > 0$ and for all p, q, we have $p + q \leq d + 1 - r$ (by hypothesis), and so $p + q \leq d$. Thus, we have $\widehat{\mathsf{HW}^p}(a) \cdot \widehat{\mathsf{HW}^q \circ F_1}(a) = 0$, which implies that either $\widehat{\mathsf{HW}^p}(a) = 0$ or $\widehat{\mathsf{HW}^q \circ F_1}(a) = 0$ for $a \neq 0$.

□

3.5 Problem Equivalent Formulation in Terms of Boolean Theory

We shall need the next lemma, which was already more or less explicit in [10].

Lemma 1. *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be any function, let q be an integer such that $0 < q < n$ and let $a \in \mathbb{F}_2^n$ be nonzero. We have $\sum_{z,m} \mathsf{HW}^{q'}(F(m) \oplus F(m \oplus z))(-1)^{a \cdot z} = 0$ for every $0 < q' \leq q$ if and only if $\widehat{b \cdot F}(a) = 0$ for every $b \in \mathbb{F}_2^n$ such that $\mathsf{HW}(b) \leq q$.*

Proof. According to [10, Eqn. (15)], we have:

$$\sum_{z,m} \mathsf{HW}^{q'}(F(m) \oplus F(m \oplus z))(-1)^{a \cdot z} = \tag{6}$$

$$\frac{1}{2^{q'}} \sum_{j=0}^{q'} \binom{q'}{j} n^{q'-j} (-1)^j \sum_{k_1 + \cdots + k_n = j} \binom{j}{k_1, \cdots, k_n} \left(\sum_{x \in \mathbb{F}_2^n} (-1)^{(\oplus_{i=1}^n k_i e_i) \cdot F(x) + a \cdot x} \right)^2 .$$

Since, for $b = \oplus_{i=1}^n k_i e_i$, we have $\sum_{x \in \mathbb{F}_2^n} (-1)^{(\oplus_{i=1}^n k_i e_i) \cdot F(x) + a \cdot x} = -2\,\widehat{b \cdot F}(a)$, the condition "$\widehat{b \cdot F}(a) = 0$ for every $b \in \mathbb{F}_2^n$ such that $\mathsf{HW}(b) \leq q$" is then clearly sufficient. Conversely, let the condition "$\widehat{b \cdot F}(a) = 0$ for every $b \in \mathbb{F}_2^n$ such that $\mathsf{HW}(b) \leq k$" be denoted by $P(k)$. We prove $P(k)$ by induction on $k \in \mathbb{N}$. $P(0)$ is clearly satisfied since $a \neq 0$. Assume that $P(k)$ is satisfied for some $0 \leq k \leq q-1$, then applying the hypothesis with $q' = k + 1$ implies that $\widehat{b \cdot F}(a) = 0$ for every b such that $\mathsf{HW}(b) = k + 1$ since we have only squares in (6) multiplied by coefficients which are all of the same sign and $P(k+1)$ is then satisfied. This completes the proof by induction. □

Incidentally, we remark that the Theorem 1 of previous Sec. 2.1 is also an immediate consequence of Lemma 1 with $F = \mathsf{Id}$.

We characterize now Eqn. (5) in terms of Fourier transform.

Proposition 2. *Let F_1 and F_2 be two permutations of $\mathbb{F}_2^n$ and d an integer smaller than n. The condition:*

$$\forall a \neq 0, \forall (p,q,r), \tag{7}$$
$$(p+q+r \leq d) \Longrightarrow \begin{cases} \widehat{\mathsf{HW}^p}(a) = 0 \ \textit{or} \\ \sum_{z,m} \mathsf{HW}^q(F_1(m) \oplus F_1(m \oplus z))(-1)^{a \cdot z} = 0 \ \textit{or} \\ \sum_{z,m} \mathsf{HW}^r(F_2(m) \oplus F_2(m \oplus z))(-1)^{a \cdot z} = 0 \ . \end{cases}$$

is satisfied if and only if:

$$\forall a \in \mathbb{F}_2^n, a \neq 0, \exists q, r / \begin{cases} \mathsf{HW}(a) + q + r = d - 1, \\ \forall b \in \mathbb{F}_2^n, \mathsf{HW}(b) \leq q \Longrightarrow \widehat{b \cdot F_1}(a) = 0, \\ \forall c \in \mathbb{F}_2^n, \mathsf{HW}(c) \leq r \Longrightarrow \widehat{c \cdot F_2}(a) = 0. \end{cases} \tag{8}$$

Proof. Condition (7) is satisfied for every (p,q,r) such that $p+q+r \leq d$ if and only if it is satisfied when p is minimal such that $\widehat{\mathsf{HW}^p}(a) \neq 0$, r is minimal such that $\sum_{z,m} \mathsf{HW}^r(F_2(m) \oplus F_2(m \oplus z))(-1)^{a \cdot z} \neq 0$ and $p+q+r \leq d$. We know that the minimum value of p such that $\widehat{\mathsf{HW}^p}(a) \neq 0$ equals $\mathsf{HW}(a)$. Let r be the minimal element defined above. Condition (7) implies then:

$$\forall q \leq d - \mathsf{HW}(a) - r, \quad \sum_{z,m} \mathsf{HW}^q(F_1(m) \oplus F_1(m \oplus z))(-1)^{a \cdot z} = 0.$$

According to Lemma 1, this latter condition is equivalent to $\forall b, \mathsf{HW}(b) \leq d - \mathsf{HW}(a) - r \Longrightarrow \widehat{b \cdot F_1}(a) = 0$ and we obtain the condition:

$$\forall a \neq 0, \exists r / \begin{cases} \forall b, \mathsf{HW}(b) \leq d - \mathsf{HW}(a) - r \Longrightarrow \widehat{b \cdot F_1}(a) = 0, \\ \forall c, \mathsf{HW}(c) < r \qquad\qquad\quad \Longrightarrow \widehat{c \cdot F_2}(a) = 0. \end{cases}$$

Now, let us replace r by $r' \doteq r - 1$. Thus $\mathsf{HW}(c) < r$ is equivalent to $\mathsf{HW}(c) \leq r'$, and condition $\mathsf{HW}(a) + q + r = d$ is equivalent to $\mathsf{HW}(a) + q + r' = d - 1$. This shows that Eqn. (8) is necessary. Clearly it is also sufficient. □

It is clear from Proposition 2 that any choice of F_2 allows to increase by one the resistance order provided by F_1 (this has already been mentioned in Sec. 3.4). Indeed, let us denote by d_1 the maximal order of resistance of F_1 in the one mask situation. Then, $\forall a \neq 0$, $\forall p, q$, $p+q \leq d_1 \Longrightarrow \widehat{b \cdot F_1}(a) = 0$. By reference to Eqn. (8), for a given $a \neq 0$, we choose:

- $q = d_1 - \mathsf{HW}(a)$, thus $\forall b \in \mathbb{F}_2^n, \mathsf{HW}(b) \leq q \Longrightarrow \widehat{b \cdot F_1}(a) = 0$ (by definition of d_1).

- $r = 0$, thus $\forall c \in \mathbb{F}_2^n, \mathsf{HW}(c) \leq r \Longrightarrow \widehat{c \cdot F_2}(a) = 0$ (indeed, $c = 0$, hence $\widehat{c \cdot F_2}(a) = \delta(a) = 0$ since $a \neq 0$).

Consequently, Eqn. (8) is met with $d = d_1 + 1$.

So, one strategy can be to start from F_1, the optimal solution with one mask (this solution is known from [10]), and then to choose F_2 so as to increase as much as possible the resistance degree. Another strategy is to find F_1 and F_2 jointly. This problem seems not to be a classical one in the general case. In the next section, we show however that it becomes a problem of coding theory when F_1 and F_2 are linear.

3.6 Solutions when F_1 and F_2 are Linear

In this section, F_1 and F_2 are assumed to be linear. For every $b, x \in \mathbb{F}_2^n$, we have $b \cdot F_1(x) = F_1^t(b) \cdot x$, where F_1^t is the so-called adjoint operator of F_1, that is, the linear mapping whose matrix is the transpose of the matrix of F_1. Then, for every nonzero $a \in \mathbb{F}_2^n$, we have $\widehat{b \cdot F_1}(a) = -\frac{1}{2}\sum_{x \in \mathbb{F}_2^n}(-1)^{(F_1^t(b) \oplus a) \cdot x}$, which equals $-2^{n-1} \neq 0$ if $F_1^t(b) = a$ and is null otherwise. Let us denote by L_1 (resp. L_2) the inverse of mapping F_1^t (resp. F_2^t). Then $\widehat{b \cdot F_1}(a)$ (resp. $\widehat{c \cdot F_2}(a)$) equals $-2^{n-1} \neq 0$ if $b = L_1(a)$ (resp. if $c = L_2(a)$) and is null otherwise.

Let also $a \neq 0$. From Eqn. (8) of Proposition 2, we can choose:

- $q = \mathsf{HW}(L_1(a)) - 1$ and
- $r = \mathsf{HW}(L_2(a)) - 1$.

Thus $d = \min\{\mathsf{HW}(a) + \mathsf{HW}(L_1(a)) + \mathsf{HW}(L_2(a)) - 1; a \neq 0\}$, which is exactly the minimal distance of the code $\{(x, L_1^t(x), L_2^t(x)); x \in \mathbb{F}_2^n\}$ (of rate 1/3 and with three disjoint information sets) minus the number 1.

Example for Linear F_1 and F_2 for $n = 8$ The optimal linear codes of length $8 \times 3 = 24$ and of dimension 8 have minimal distance 8. For instance, code $[24, 8, 8]$ is a subcode of code $[24, 12, 8]$, that is itself the extension of the quadratic-residue (QR) code of length 23.

By properly arranging the bits in the codewords, the generator matrix can write $\begin{pmatrix} I_8 & L_1^t & L_2^t \end{pmatrix}$, where:

$$L_1^t = \begin{pmatrix} 1&0&0&0&0&1&0&1 \\ 1&0&0&0&0&1&1&1 \\ 1&0&1&1&1&0&0&1 \\ 1&0&1&1&1&0&1&0 \\ 1&0&1&1&1&1&1&0 \\ 0&1&1&0&0&1&1&1 \\ 0&1&0&1&0&1&1&1 \\ 0&1&0&0&1&0&0&0 \end{pmatrix}, \quad L_2^t = \begin{pmatrix} 0&1&1&0&0&1&1&0 \\ 0&0&0&1&1&0&1&0 \\ 1&1&1&1&0&1&1&1 \\ 1&0&1&0&0&0&0&1 \\ 1&1&0&1&1&0&1&0 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&1&1&0 \\ 1&0&0&0&1&1&1&1 \end{pmatrix}. \tag{9}$$

Those matrices are of full rank, namely 8, and their inverses are:

$$(L_1^t)^{-1} = (L_1^{-1})^t = \begin{pmatrix} 0&1&1&0&1&0&0&0\\ 1&0&1&1&1&1&1&1\\ 1&0&0&1&0&0&1&1\\ 1&0&0&1&0&1&0&1\\ 1&0&1&1&1&1&1&0\\ 0&0&0&1&1&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0 \end{pmatrix}, \quad (L_2^t)^{-1} = (L_2^{-1})^t = \begin{pmatrix} 0&0&0&0&0&1&1&1\\ 0&1&0&0&1&1&1&1\\ 1&1&1&0&1&0&1&1\\ 0&1&0&1&1&0&1&1\\ 1&1&0&1&1&1&1&0\\ 1&1&1&0&0&0&0&1\\ 1&1&0&0&0&1&0&1\\ 1&1&1&1&1&1&0&0 \end{pmatrix}.$$

The technique to find those matrices is described in Appendix A.

We note that the binary linear code $\{(x, F_1(x)); x \in \mathbb{F}_2^8\}$ has minimal distance 3, and that the binary linear code $\{(x, F_2(x)); x \in \mathbb{F}_2^8\}$ has minimal distance 4. So those two codes are non-optimal, because the best linear code of length 16 and dimension 8 has minimal distance 5 [11, Tab. 1 in §4.1]. This noting justifies that it is indeed relevant to search for the bijections doublet (F_1, F_2) together instead of one after the other, independently. It also suggests that non-linear codes might still achieve better.

Example for Linear F_1 and F_2 for $n = 4$ It is also possible to construct a rate 1/3 linear code of dimension 4 with three distinct information sets, which is suitable to protect DES. This solution generates two bijections, given in Eqn. (12), that jointly allow to resist high-order attacks of order 1 to 5 inclusive. The detail of the construction is given in Appendix B.

Security Validation An implementation with a leakage function $\mathscr{L}$ is vulnerable at order d if $\mathbb{E}[(\mathscr{L} - \mathbb{E}[\mathscr{L}])^d \mid Z = z]$ depends on z, *i.e.* if the variance of this random variable is strictly positive ($\mathsf{Var}[\mathbb{E}[(\mathscr{L} - \mathbb{E}[\mathscr{L}])^d \mid Z]] > 0$). In this case, the asymptotic HO-CPA correlation coefficient $\rho_{\mathrm{opt}}^{(d)}$, equal to [20, Eqn. (15) at page 802]:

$$\rho_{\mathrm{opt}}^{(d)} \doteq \sqrt{\frac{\mathsf{Var}[\mathbb{E}[(\mathscr{L} - \mathbb{E}[\mathscr{L}])^d \mid Z]]}{\mathsf{Var}[(\mathscr{L} - \mathbb{E}[\mathscr{L}])^d]}}\ , \tag{10}$$

is non-zero. The table 1 gives the values of $\rho_{\mathrm{opt}}^{(d)}$ for the second-order Boolean masking of bytes ($n = 8$), without (at left hand-side) and with (at right hand-side) leakage squeezing. In the middle of Tab. 1, an intermediate case is shown: it corresponds to a "partial" leakage squeezing, where a bijection is applied only on one mask out of the two. We notice that the simulation results of partial leakage squeezing are in line with the theoretical analysis carried out in Sec. 3.4: the order of resistance is indeed incremented by one. The two variances involved in Eqn. (10) were computed using a multiprecision integer library; therefore, when $\rho_{\mathrm{opt}}^{(d)}$ is reported as 0 (integer zero, not the approximated floating number 0.000000), we really mean that $\mathbb{E}[(\mathscr{L} - \mathbb{E}[\mathscr{L}])^d \mid Z = z]$ does not depend on z.

For the sake of comparison, we also report in this table the results obtained with one mask. In such case, both the best linear and non-linear squeezing bijection F can be characterized. It is relevant to consider the linear bijections F

as they allow an efficient protection against HO-CPA, whether the device leaks in Hamming weight or distance. The best linear F for leakage squeezing with one mask is secure against attacks of orders up to 4. It can be used with two masks, thereby granting a security up to order $4 + 1 = 5$. Our results, that are not based on the extension of a single mask solution, is a security against HO-CPA of orders up to 7. Therefore, our method provides a free advantage of two orders. Now, with one mask, the best achievable security is gotten by the use of a non-linear F. This function does only protect against attacks that exploit the Hamming distance (and not the Hamming weight), but allows to reach a resistance up to HO-CPA of order 5. here also, our linear solution with two masks is better than merely this code used with one mask extended with another mask: it protected at order up to $7 > 5 + 1$. Besides, it is interesting to compare the first nonzero correlation coefficients with and without leakage squeezing:

- with one mask, $\rho_{\text{opt}}^{(d=2)}(\text{no LS})/\rho_{\text{opt}}^{(d=5)}(\text{LS}) = 0.258199/0.023258 \approx 11$, and
- with two masks, $\rho_{\text{opt}}^{(d=3)}(\text{no LS})/\rho_{\text{opt}}^{(d=8)}(\text{LS}) = 0.038886/0.000446 \approx 87$.

So, in front of leakage squeezing, not only the attacker shall conduct an attack of much higher order, but also she will get a very degraded distinguisher value.

On $n = 4$ bits, the optimal first-order leakage squeezing is linear and allows to reach resistance order of 3. The used optimal code is $[8, 4, 4]$. For the second-order leakage squeezing, we can resort to the linear code $[12, 4, 6]$, that improves by two $(6 - 4 = 2)$ orders (with only one additional mask) the resistance against HO-CPA. By the trivial construct of Sec. 3.4, only one additional order of resistance would have been gained. A summary of the results is shown in Tab. 2. The improvement from the "straightforward" to the "squeezed" masking is of two orders with one mask and three orders with two masks.

4 Conclusions and Perspectives

Leakage squeezing has been thoroughly studied in [10] in the context where one sole mask is used. This paper investigates the potential of leakage squeezing extension to second-order leakage squeezing, *i.e.* using two independent masks instead of only one. Trivially, the addition of one mask increases the resistance against HO-CPA by one order. Our analysis allows to characterize (in Proposition 2) the conditions to reach higher resistance. The optimal solutions are not as easy to find as in the case with one mask. Nonetheless, for the special case of linear bijections, we find that one solution (probably not optimal) consists in finding a rate 1/3 linear code of maximal minimal distance with three disjoint information sets. The optimal $[24, 8, 8]$ linear code fulfills this condition, and makes it possible to resist attacks of all orders from 1 to 7 included. Concretely speaking, this result means that the same security level as a 7th-order attack is attainable with 2 instead of 7 masks, thus at a much lower implementation cost.

Finding better, for instance non-linear, bijections, could allow to further improve on top of these results. In particular, a thorough study of rate 1/2 codes

Table 1. Optimal 2O-CPA correlation coefficient for zero-offset attacks at order d, without and with leakage squeezing (LS) on $n = 8$ bits. Results — that are normalized, because the absolute value of a Pearson correlation coefficient lives in $[0, 1]$ — are rounded at the sixth decimal.

	One mask (see [10])			**Two masks (this paper, *i.e.* [9])**			
Order d	**Without LS**	**With LS**		**Without LS**	**With "partial" LS**		**With LS**
	$F = \mathsf{Id}$	**Optimal linear ([10, App. B])**	**Optimal non-linear ([10, Sec. 4.2])**	$F_1 = F_2 = \mathsf{Id}$	$F_1 = \mathsf{Id}$, **but** F_2 **as [10, App. B]**	$F_2 = \mathsf{Id}$, **but** F_2 **as [10, Sec. 4.2]**	**(Non-optimal) linear (*cf* Sec. 3.6)**
1	0	0	0	0	0	0	0
2	0.258199	0	0	0	0	0	0
3	0	0	0	0.038886	0	0	0
4	0.235341	0	0	0	0	0	0
5	0	0.023231	0	0.049669	0	0	0
6	0.197908	0.016173	0.023258	0.003403	0.001286	0	0
7	0	0.042217	0	0.045585	0.000868	0.000726	0
8	0.164595	0.032796	0.046721	0.006820	0.002644	0.000682	0.000446

Table 2. Optimal 2O-CPA correlation coefficient for zero-offset attacks at order d, without and with leakage squeezing (LS) on $n = 4$ bits. Results — that are normalized, because the absolute value of a Pearson correlation coefficient lives in $[0, 1]$ — are rounded at the sixth decimal.

	One mask (see [10])		**Two masks (this paper, *i.e.* [9])**		
Order d	**Without LS**	**With LS**	**Without LS**	**With "partial" LS**	**With LS**
	$F = \mathsf{Id}$	**Optimal linear ([8, Eqn. (13)])**	$F_1 = F_2 = \mathsf{Id}$	$F_1 = \mathsf{Id}$, **and** $F_2 \neq \mathsf{Id}$ **is [8, Eqn. (13)]**	**(Non-optimal) linear (*cf* Sec. 3.6)**
1	0	0	0	0	0
2	0.377964	0	0	0	0
3	0	0	0.081289	0	0
4	0.363815	0.191663	0	0	0
5	0	0	0.105175	0.021035	0
6	0.346246	0.283546	0.015973	0.015973	0.022590

with two complementary information sets exists [1]. However, such work is missing in general for rate $1/d$ codes with $d > 2$ distinct information sets.

Another perspective is to integrate the second-order leakage squeezing with "hyperpipelined" designs [16], "threshold implementations" [18] or "multi-party computation" [21] masking schemes, so as to improve their order of resistance while at the same time removing the latent leakage by glitches (if the logic is not concealed in memories).

References

1. Claude Carlet, Philippe Gaborit, Jon-Lark Kim, and Patrick Solé. A new class of codes for Boolean masking of cryptographic computations, October 6 2011. http://arxiv.org/abs/1110.1193. To appear in IEEE Transactions on Information Theory.
2. Claude Carlet, Philippe Gaborit, Jon-Lark Kim, and Patrick Solé. A New Class of Codes for Boolean Masking of Cryptographic Computations. *IEEE Transactions on Information Theory*, 58(9):6000–6011, 2012.
3. Markus Dichtl. A new method of black box power analysis and a fast algorithm for optimal key search. *J. Cryptographic Engineering*, 1(4):255–264, 2011.
4. DPA Contest (2^{nd} edition), 2009–2010. http://www.DPAcontest.org/v2/.
5. Markus Grassl. Bounds on the minimum distance of linear codes and quantum codes. Online available at http://www.codetables.de/, 2007. Accessed on 2012-07-23.
6. Annelie Heuser, Werner Schindler, and Marc Stöttinger. Revealing side-channel issues of complex circuits by enhanced leakage models. In Wolfgang Rosenstiel and Lothar Thiele, editors, *DATE*, pages 1179–1184. IEEE, 2012.
7. Yang Li, Daisuke Nakatsu, Qi Li, Kazuo Ohta, and Kazuo Sakiyama. Clockwise Collision Analysis – Overlooked Side-Channel Leakage Inside Your Measurements. Cryptology ePrint Archive, Report 2011/579, october 2011. http://eprint.iacr.org/2011/579.
8. Houssem Maghebi, Sylvain Guilley, Claude Carlet, and Jean-Luc Danger. Classification of High-Order Boolean Masking Schemes and Improvements of their Efficiency. Cryptology ePrint Archive, Report 2011/520, September 2011. http://eprint.iacr.org/2011/520.
9. Houssem Maghrebi, Claude Carlet, Sylvain Guilley, and Jean-Luc Danger. Leakage Squeezing of Order Two. In *INDOCRYPT*, volume 7668 of *LNCS*, pages 120–139. Springer, December 9-12 2012. Kolkata, India.
10. Houssem Maghrebi, Claude Carlet, Sylvain Guilley, and Jean-Luc Danger. Optimal first-order masking with linear and non-linear bijections. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *AFRICACRYPT*, volume 7374 of *Lecture Notes in Computer Science*, pages 360–377. Springer, 2012.
11. Houssem Maghrebi, Sylvain Guilley, Claude Carlet, and Jean-Luc Danger. Optimal First-Order Masking with Linear and Non-Linear Bijections. In *AFRICACRYPT*, LNCS. Springer, July 10-12 2012. Al Akhawayn University in Ifrane, Morocco.
12. Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage Squeezing Countermeasure Against High-Order Attacks. In *WISTP*, volume 6633 of *LNCS*, pages 208–223. Springer, June 1-3 2011. Heraklion, Greece. DOI: 10.1007/978-3-642-21040-2_14.

13. 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. San Francisco, CA, USA.
14. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer, December 2006. ISBN 0-387-30857-1, http://www.dpabook.org/.
15. Stefan Mangard and Kai Schramm. Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In *CHES*, volume 4249 of *LNCS*, pages 76–90. Springer, October 10-13 2006. Yokohama, Japan.
16. Amir Moradi and Oliver Mischke. Glitch-free Implementation of Masking in Modern FPGAs. In *HOST*, IEEE Computer Society, pages 89–95, June 2-3 2012. Moscone Center, San Francisco, CA, USA. DOI: 10.1109/HST.2012.6224326.
17. Amir Moradi and Oliver Mischke. How Far Should Theory be from Practice? Evaluation of a Countermeasure. In *CHES*, September 9-12 2012. Leuven, Belgium.
18. Svetla Nikova, Vincent Rijmen, and Martin Schläffer. Secure hardware implementation of nonlinear functions in the presence of glitches. *J. Cryptology*, 24(2):292–321, 2011.
19. Bart Preneel and Tsuyoshi Takagi, editors. *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 – October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*. Springer, 2011.
20. Emmanuel Prouff, Matthieu Rivain, and Régis Bevan. Statistical Analysis of Second Order Differential Power Analysis. *IEEE Trans. Computers*, 58(6):799–811, 2009.
21. Emmanuel Prouff and Thomas Roche. Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. In Preneel and Takagi [19], pages 63–78.
22. Mathieu Renauld, François-Xavier Standaert, Nicolas Veyrat-Charvillon, Dina Kamel, and Denis Flandre. A Formal Study of Power Variability Issues and Side-Channel Attacks for Nanoscale Devices. In *EUROCRYPT*, volume 6632 of *LNCS*, pages 109–128. Springer, May 2011. Tallinn, Estonia.
23. Matthieu Rivain and Emmanuel Prouff. Provably Secure Higher-Order Masking of AES. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *LNCS*, pages 413–427. Springer, 2010.
24. Werner Schindler, Kerstin Lemke, and Christof Paar. A Stochastic Model for Differential Side Channel Cryptanalysis. In LNCS, editor, *CHES*, volume 3659 of *LNCS*, pages 30–46. Springer, Sept 2005. Edinburgh, Scotland, UK.
25. Shaunak Shah, Rajesh Velegalati, Jens-Peter Kaps, and David Hwang. Investigation of DPA Resistance of Block RAMs in Cryptographic Implementations on FPGAs. In Viktor K. Prasanna, Jürgen Becker, and René Cumplido, editors, *ReConFig*, pages 274–279. IEEE Computer Society, 2010.
26. Kris Tiri, Davis Hwang, Alireza Hodjat, Bo-Cheng Lai, Shenglin Yang, Patrick Schaumont, and Ingrid Verbauwhede. Prototype IC with WDDL and Differential Routing – DPA Resistance Assessment. In LNCS, editor, *Proceedings of CHES'05*, volume 3659 of *LNCS*, pages 354–365. Springer, August 29 – September 1 2005. Edinburgh, Scotland, UK.
27. Kris Tiri and Ingrid Verbauwhede. A VLSI Design Flow for Secure Side-Channel Attack Resistant ICs. In *DATE*, pages 58–63. IEEE Computer Society, 2005. http://dx.doi.org/10.1109/DATE.2005.44.

28. Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal Key Enumeration Algorithm and its Application to Side-Channel Attacks. Cryptology ePrint Archive, Report 2011/610, 2011. http://eprint.iacr.org/2011/610/.
29. Jason Waddle and David Wagner. Towards Efficient Second-Order Power Analysis. In *CHES*, volume 3156 of *LNCS*, pages 1–15. Springer, 2004. Cambridge, MA, USA.

A Isolation of Three Information Sets from Code $[24, 8, 8]$

If F_1 and F_2 are two linear bijections then the linear code $\{(x, F_1(x), F_2(x)); x \in \mathbb{F}_2^n\}$ has $\{1, \cdots, n\}$, $\{n+1, \cdots, 2n\}$ and $\{2n+1, \cdots, 3n\}$ for information sets, since the restriction of the generator matrix of this code to the columns indexed in each of these three sets is invertible. Conversely, if a $[3n, n, d]$ code C is known with three disjoint information sets, then after rearranging the columns of its generator matrix so that these three information sets are $\{1, \cdots, n\}$, $\{n+1, \cdots, 2n\}$ and $\{2n+1, \cdots, 3n\}$, we have $C = \{(\phi_0(x), \phi_1(x), \phi_2(x)); x \in \mathbb{F}_2^n\}$ where ϕ_0, ϕ_1 and ϕ_2 are bijective. Then, by trading the dummy variable x for $y = \phi_0(x)$ through one-to-one function ϕ_0, we get $C = \{(y, \phi_1 \circ \phi_0^{-1}(y), \phi_2 \circ \phi_0^{-1}(y)); y \in \mathbb{F}_2^n\}$ and we can take $F_1 = \phi_1 \circ \phi_0^{-1}$ and $F_2 = \phi_2 \circ \phi_0^{-1}$.

One generator matrix for the $[24, 8, 8]$ code can be obtained as a submatrix of extended QR-code of length 23[1], such as:

$$
\begin{array}{cccccccccccccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\left(\begin{array}{cccccccccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1
\end{array}\right)
\end{array}.
$$

The goal is to rearrange the columns of this matrix to get a form:

$$\begin{pmatrix} M_0^t & M_1^t & M_2^t \end{pmatrix}, \tag{11}$$

where M_0, M_1 and M_2 are 8×8 invertible matrices with elements in $\mathbb{F}_2$. The research algorithm is as follows: first, an invertible 8×8 matrix (M_0) is searched. There are $\binom{24}{8} = 735,471$ of them[2]. We find one M_0^t by considering the columns $[\![2, 9]\!]$. Second, the $\binom{16}{8} = 12,870$ permutations of columns $\{1\} \cup [\![10, 24]\!]$ are tested for a partitioning into two invertible matrices $\begin{pmatrix} M_1^t & M_2^t \end{pmatrix}$. For instance, M_1^t can be the columns $\{1, 10, 11, 12, 13, 15, 17, 18\}$ and M_2^t the columns $\{14, 16\} \cup$

[1] See: http://www.mathe2.uni-bayreuth.de/cgi-bin/axel/codedb?extensioncodeid+39649+2+8 [5].

[2] This amount of tries is still manageable on a standard desktop personal computer; all the more so as, in practice, we find very quickly a solution as the number of partitionings that yield an invertible 8×8 matrix is $310,400$ (which represents around 42% of the possible partitionings).

$[\![19, 24]\!]$. Those define the three bijections $\phi_i : \mathbb{F}_2^8 \to \mathbb{F}_2^8, x \mapsto M_i \times x$, for $i \in \{0, 1, 2\}$. After that, we get a generating matrix in systematic form $\begin{pmatrix} I_8 & L_1^t & L_2^t \end{pmatrix}$; The matrices L_1^t and L_2^t are defined as $L_1^t = M_1 \times M_0^{-1} = \left((M_0^t)^{-1} \times M_1^t\right)^t$ and $L_2^t = M_2 \times M_0^{-1} = \left((M_0^t)^{-1} \times M_2^t\right)^t$, and their values are given in Eqn. (9).

The resulting functions F_1 and F_2 are tabulated in Tab. 3.

A priori, it was not clear whether or not the $[24, 8, 8]$ code could be cut into three disjoint information sets. However, in this case, it is, as just described, and in a non-unique way. For instance, the same shape as Eqn. (11) can be obtained by selecting for M_0^t the columns of index 0 modulo 3, for M_1^t the columns of index 1 modulo 3, and for M_2^t the columns of index 2 modulo 3. This partitioning is not equivalent to the previous one, as the columns for the new matrices pick up columns from all three previous ones. However, results in terms of correlation coefficient (*c.f.* Eqn. (10)) are the same.

B Isolation of Three Information Sets from Code $[12, 4, 6]$

The same research algorithm can be used for the optimal code of length 12 and dimension 4, *i.e.* $[12, 4, 6]$. Its minimal distance is 6. One generator matrix for the $[12, 4, 6]$ code is[3]:

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{matrix}$$
$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

By gathering columns $\{1, 4, 7, 2\}$ as M_0^t, columns $\{10, 5, 8, 3\}$ as M_1^t and columns $\{11, 6, 9, 12\}$ as M_2^t, the code rewrites in the form of Eqn. (11) where M_0, M_1 and M_2 are invertible. Specifically, we have:

$$M_0^t = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad M_1^t = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \quad M_2^t = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix};$$
$$(M_0^t)^{-1} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \quad (M_1^t)^{-1} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad (M_2^t)^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

[3] See: http://www.math.colostate.edu/~betten/research/codes/BOUNDS/bounds_GF2.html.

So, by seeing x as a column $x \doteq \begin{pmatrix} x_{n-1} \\ \vdots \\ x_0 \end{pmatrix}$, we also define:

$$F_1(x) \doteq \left[(M_1^t)^{-1} \times M_0^t\right] \times x = \begin{pmatrix} 1\,0\,0\,1 \\ 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \end{pmatrix} \times x \ ,$$

$$F_2(x) \doteq \left[(M_2^t)^{-1} \times M_0^t\right] \times x = \begin{pmatrix} 0\,1\,0\,1 \\ 0\,0\,1\,1 \\ 1\,1\,1\,1 \\ 1\,0\,0\,0 \end{pmatrix} \times x \ . \tag{12}$$

The resulting functions F_1 and F_2 are tabulated in Tab. 3.

Table 3. Truth table for the found linear bijections F_1 and F_2 of $\mathbb{F}_2^8$ (resp. of $\mathbb{F}_2^4$) in the *top* (resp. in the *bottom*), that provide resistance up to level 7 (resp. 5).

$\{F_1(x); x \in \mathbb{F}_2^8\} =$	$\{F_2(x); x \in \mathbb{F}_2^8\} =$
{ 0x00, 0x70, 0x68, 0x18, 0x58, 0x28, 0x30, 0x40, 0xcc, 0xbc, 0xa4, 0xd4, 0x94, 0xe4, 0xfc, 0x8c, 0x7d, 0x0d, 0x15, 0x65, 0x25, 0x55, 0x4d, 0x3d, 0xb1, 0xc1, 0xd9, 0xa9, 0xe9, 0x99, 0x81, 0xf1, 0xc9, 0xb9, 0xa1, 0xd1, 0x91, 0xe1, 0xf9, 0x89, 0x05, 0x75, 0x6d, 0x1d, 0x5d, 0x2d, 0x35, 0x45, 0xb4, 0xc4, 0xdc, 0xac, 0xec, 0x9c, 0x84, 0xf4, 0x78, 0x08, 0x10, 0x60, 0x20, 0x50, 0x48, 0x38, 0x83, 0xf3, 0xeb, 0x9b, 0xdb, 0xab, 0xb3, 0xc3, 0x4f, 0x3f, 0x27, 0x57, 0x17, 0x67, 0x7f, 0x0f, 0xfe, 0x8e, 0x96, 0xe6, 0xa6, 0xd6, 0xce, 0xbe, 0x32, 0x42, 0x5a, 0x2a, 0x6a, 0x1a, 0x02, 0x72, 0x4a, 0x3a, 0x22, 0x52, 0x12, 0x62, 0x7a, 0x0a, 0x86, 0xf6, 0xee, 0x9e, 0xde, 0xae, 0xb6, 0xc6, 0x37, 0x47, 0x5f, 0x2f, 0x6f, 0x1f, 0x07, 0x77, 0xfb, 0x8b, 0x93, 0xe3, 0xa3, 0xd3, 0xcb, 0xbb, 0x7b, 0x0b, 0x13, 0x63, 0x23, 0x53, 0x4b, 0x3b, 0xb7, 0xc7, 0xdf, 0xaf, 0xef, 0x9f, 0x87, 0xf7, 0x06, 0x76, 0x6e, 0x1e, 0x5e, 0x2e, 0x36, 0x46, 0xca, 0xba, 0xa2, 0xd2, 0x92, 0xe2, 0xfa, 0x8a, 0xb2, 0xc2, 0xda, 0xaa, 0xea, 0x9a, 0x82, 0xf2, 0x7e, 0x0e, 0x16, 0x66, 0x26, 0x56, 0x4e, 0x3e, 0xcf, 0xbf, 0xa7, 0xd7, 0x97, 0xe7, 0xff, 0x8f, 0x03, 0x73, 0x6b, 0x1b, 0x5b, 0x2b, 0x33, 0x43, 0xf8, 0x88, 0x90, 0xe0, 0xa0, 0xd0, 0xc8, 0xb8, 0x34, 0x44, 0x5c, 0x2c, 0x6c, 0x1c, 0x04, 0x74, 0x85, 0xf5, 0xed, 0x9d, 0xdd, 0xad, 0xb5, 0xc5, 0x49, 0x39, 0x21, 0x51, 0x11, 0x61, 0x79, 0x09, 0x31, 0x41, 0x59, 0x29, 0x69, 0x19, 0x01, 0x71, 0xfd, 0x8d, 0x95, 0xe5, 0xa5, 0xd5, 0xcd, 0xbd, 0x4c, 0x3c, 0x24, 0x54, 0x14, 0x64, 0x7c, 0x0c, 0x80, 0xf0, 0xe8, 0x98, 0xd8, 0xa8, 0xb0, 0xc0 },	{ 0x00, 0xf6, 0xf8, 0x0e, 0xcb, 0x3d, 0x33, 0xc5, 0x79, 0x8f, 0x81, 0x77, 0xb2, 0x44, 0x4a, 0xbc, 0x19, 0xef, 0xe1, 0x17, 0xd2, 0x24, 0x2a, 0xdc, 0x60, 0x96, 0x98, 0x6e, 0xab, 0x5d, 0x53, 0xa5, 0x25, 0xd3, 0xdd, 0x2b, 0xee, 0x18, 0x16, 0xe0, 0x5c, 0xaa, 0xa4, 0x52, 0x97, 0x61, 0x6f, 0x99, 0x3c, 0xca, 0xc4, 0x32, 0xf7, 0x01, 0x0f, 0xf9, 0x45, 0xb3, 0xbd, 0x4b, 0x8e, 0x78, 0x76, 0x80, 0x7f, 0x89, 0x87, 0x71, 0xb4, 0x42, 0x4c, 0xba, 0x06, 0xf0, 0xfe, 0x08, 0xcd, 0x3b, 0x35, 0xc3, 0x66, 0x90, 0x9e, 0x68, 0xad, 0x5b, 0x55, 0xa3, 0x1f, 0xe9, 0xe7, 0x11, 0xd4, 0x22, 0x2c, 0xda, 0x5a, 0xac, 0xa2, 0x54, 0x91, 0x67, 0x69, 0x9f, 0x23, 0xd5, 0xdb, 0x2d, 0xe8, 0x1e, 0x10, 0xe6, 0x43, 0xb5, 0xbb, 0x4d, 0x88, 0x7e, 0x70, 0x86, 0x3a, 0xcc, 0xc2, 0x34, 0xf1, 0x07, 0x09, 0xff, 0x2f, 0xd9, 0xd7, 0x21, 0xe4, 0x12, 0x1c, 0xea, 0x56, 0xa0, 0xae, 0x58, 0x9d, 0x6b, 0x65, 0x93, 0x36, 0xc0, 0xce, 0x38, 0xfd, 0x0b, 0x05, 0xf3, 0x4f, 0xb9, 0xb7, 0x41, 0x84, 0x72, 0x7c, 0x8a, 0x0a, 0xfc, 0xf2, 0x04, 0xc1, 0x37, 0x39, 0xcf, 0x73, 0x85, 0x8b, 0x7d, 0xb8, 0x4e, 0x40, 0xb6, 0x13, 0xe5, 0xeb, 0x1d, 0xd8, 0x2e, 0x20, 0xd6, 0x6a, 0x9c, 0x92, 0x64, 0xa1, 0x57, 0x59, 0xaf, 0x50, 0xa6, 0xa8, 0x5e, 0x9b, 0x6d, 0x63, 0x95, 0x29, 0xdf, 0xd1, 0x27, 0xe2, 0x14, 0x1a, 0xec, 0x49, 0xbf, 0xb1, 0x47, 0x82, 0x74, 0x7a, 0x8c, 0x30, 0xc6, 0xc8, 0x3e, 0xfb, 0x0d, 0x03, 0xf5, 0x75, 0x83, 0x8d, 0x7b, 0xbe, 0x48, 0x46, 0xb0, 0x0c, 0xfa, 0xf4, 0x02, 0xc7, 0x31, 0x3f, 0xc9, 0x6c, 0x9a, 0x94, 0x62, 0xa7, 0x51, 0x5f, 0xa9, 0x15, 0xe3, 0xed, 0x1b, 0xde, 0x28, 0x26, 0xd0 }.

$\{F_1(x); x \in \mathbb{F}_2^4\} =$	$\{F_2(x); x \in \mathbb{F}_2^4\} =$
{ 0x0, 0xf, 0x6, 0x9, 0x5, 0xa, 0x3, 0xc, 0xb, 0x4, 0xd, 0x2, 0xe, 0x1, 0x8, 0x7 },	{ 0x0, 0xe, 0x6, 0x8, 0xa, 0x4, 0xc, 0x2, 0x3, 0xd, 0x5, 0xb, 0x9, 0x7, 0xf, 0x1 }.