EPiC Series in Computing

Volume 87, 2022, Pages 83–99

Proceedings of 10th International Workshop
on Security Proofs for Embedded Systems

Towards Finding Best Linear Codes for Side-Channel Protections

Wei Cheng[1], Yi Liu[1], Sylvain Guilley[2,1], and Olivier Rioul[1]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120, Palaiseau, France,
firstname.lastname@telecom-paris.fr

[2] Secure-IC S.A.S., 75015, Paris, France, sylvain.guilley@secure-ic.com

Abstract

Side-channel attacks aim at extracting secret keys from cryptographic devices. Randomly masking the implementation is a provable way to protect the secrets against this threat. Recently, various masking schemes have converged to the "code-based masking" philosophy. In code-based masking, different codes allow for different levels of side-channel security. In practice, for a given leakage function, it is important to select the code which enables the best resistance, i.e., which forces the attacker to capture and analyze the largest number of side-channel traces.

This paper is a first attempt to address the constructive selection of the optimal codes in the context of side-channel countermeasures, in particular for code-based masking when the device leaks information in the Hamming weight leakage model. We show that the problem is related to the weight enumeration of the extended dual of the masking code. We first present mathematical tools to study those weight enumeration polynomials, and then provide an efficient method to search for good codes, based on a lexicographic sorting of the weight enumeration polynomial from lowest to highest degrees.

Keywords. Side-Channel Analysis, Masking Scheme, Information-Theoretic Metric, Linear Code, Security Formalization, Weight Distribution.

1 Introduction

Cryptographic devices are prone to side-channel attacks. These attacks consist in the analysis of unintentional leakages, arising from within the computation of the cryptographic algorithms. Leakages are captured as execution traces by fast sampling apparatus, such as high bandwidth oscilloscopes. In a typical side-channel attack, numerous traces are gathered into a dataset, referred to as an acquisition campaign. In the recent years, strong efforts have been deployed for devising techniques to extract as much information as possible about the secret key. Up-to-date exploits concern template attacks, including machine learning and artificial intelligence attacks.

It is thus extremely important to ensure some reliable protection against those attacks. Countermeasures are optimized accordingly, favoring those whose implementation is mathematically provable. For this reason, random masking [12, 20] has turned out to be the countermeasure of reference.

Recently, *general code-based masking* (GCM) [6, 24] has been promoted as a way to unite several masking schemes. The peculiarities of inner product masking, direct sum masking, etc. can indeed be united into the GCM framework. This framework is amenable to encoding algorithms employing data units as bit strings of ℓ bits—where for instance, $\ell = 8$ for AES (a byte-oriented block cipher) and $\ell = 4$ for PRESENT (a nibble-oriented block cipher). Therefore, codes in GCM are naturally built with $\mathbb{F}_{2^\ell}$ as the base field.

However, optimizing codes which underlie the GCM implementation is still an open question not fully resolved. Indeed, as of today, two leakage models co-exist:

- The *probing leakage model* (at word level, in $\mathbb{F}_{2^\ell}$);
- The *bounded moment leakage model* (at bit level, in $\mathbb{F}_2$).

Accordingly, these two leakage models are concerned with two different adversarial strategies, namely:

- The *probing* model considers an attacker who can place a limited number of probes to acquire a linear dump of the consecutive values taken on by the probed variables. This model is an extension of the one proposed in the seminal paper from Ishai, Sahai and Wagner [12] which only considered bits. Current probing models encompass probing of full-width registers [20].
- The *bounded moment* model [2] considers the realization of a (high-order) correlation analysis, whereby different signals are combined so as to weaken, or eventually cancel out completely, the effect of the mask. These attacks exploit the signals arising from any bits manipulated in the netlist, and the order of the attack is the limiting complexity factor.

Now, in the context of the practical security evaluation of a device, both models are to be considered at once. The commonality between both models is that the masking strength relates to the *dual distance* of the masking code [5, 18]. Also, the bit level security relates to the extension of the code into the base field [5, 7]. Putting everything together,

- The *probing* model is limited by the number of probes t: The masking code in $\mathbb{F}_{2^\ell}$ must have a dual distance strictly greater than t.
- The *bounded moment* model requires that the subfield extension of the masking code from $\mathbb{F}_{2^\ell}$ to $\mathbb{F}_2$ has a dual distance as high as possible. It is of course at least as large as that of the code on $\mathbb{F}_{2^\ell}$, but can (and ideally should) be strictly larger.

Essentially, two leakage models are connected with each other. Indeed, given a linear code over $\mathbb{F}_{2^\ell}$, it is always feasible to extend it into the subfield $\mathbb{F}_2$. However, this extension depends on both the irreducible polynomial used in $\mathbb{F}_{2^\ell}$ and the basis used for the extension. In this paper, we focus on the latter since the finite field is fixed for a specific cryptographic algorithm like AES or PRESENT. Furthermore, another benefit of extending codes from $\mathbb{F}_{2^\ell}$ to $\mathbb{F}_2$ is that it sets the same baseline for all linear codes over $\mathbb{F}_2$, resulting that their coding-theoretic properties can be fairly compared.

Contributions. In this paper, we show how to build codes with length $n = t + 1$ which have a good bit-level security order. We revisit the code extension from $\mathbb{F}_{2^\ell}$ to $\mathbb{F}_2$ by using subfield representation with trace-orthogonal bases, which brings the commutative relationship between subfield representation and duality of the code. Next, we connect the side-channel resistance of a code-based masking to the whole weight distribution of corresponding linear codes. With the lexicographical order of weight distribution, we show how to choose the best one among them, and validate our approach by an information-theoretic assessment. In summary, our findings empower the code-based masking by providing optimal linear codes which can maximize the side-channel resistance from an information-theoretic perspective.

2 Background

2.1 Preliminaries

We first introduce several definitions which will be used throughout this paper.

Definition 1 (Linear code parameters [15]). *A linear code C is a set of vectors, called codewords, which form a vector space over some finite field $\mathbb{F}_{2^\ell}$. The parameters of the linear code C is a triple (n, k, d), where n is the code length, k is its dimension, and d is its minimum (Hamming) distance. They are denoted by $[n, k, d]_{2^\ell}$ to refer to the field on which the code is defined.*

Definition 2 (Complement of a linear code). *Two linear codes C_1 and C_2 are complementary to one another if $C_1 \cap C_2 = \{0\}$, where 0 is the all-zero codeword.*

It is always possible to build a complement of a code C: The generating matrix $\mathcal{G}_C$ of C can be complemented by vectors (e.g., randomly, one by one) until it forms a basis of the vector space. The complemented vectors form the generating matrix of a complement code of C.

Definition 3 (Dual code [15] and dual distance). *The dual code of a code C is the linear code consisting of the set of all vectors orthogonal to all codewords of C. The dual distance $d_C^\perp = d_{C^\perp}$ of the code C is the minimum distance of its dual code $C^\perp$.*

Definition 4 (Weight distribution [15] and kissing number). *The (Hamming) weight distribution of a code C of length n is the $(n+1)$-tuple of integers A_i, $0 \leq i \leq n$, such that $A_i = \#\{c \in C, w_H(c) = i\}$ (where w_H is the Hamming weight).*

In particular, the kissing number A_d is the number of codewords at minimum distance d to any codeword.

Definition 5 (Subfield extension of a code [15]). *The subfield representation of $x \in \mathbb{F}_{2^\ell}$ is its vector of coordinates $[x] \in \mathbb{F}_2^\ell$, which depends on the choice of the basis of $\mathbb{F}_{2^\ell}$ over $\mathbb{F}_2$.*

The subfield extension $[C]$ is the set of all vectors obtained from the codewords of C by taking the subfield representation of every component.

Considering a generator matrix of a linear code C of size $k \times n$ in $\mathbb{F}_{2^\ell}$, the generator matrix of the extended code $[C]$ has a size of $k\ell \times n\ell$ in $\mathbb{F}_2$.

As demonstrated in [6,7], a linear code is all the better (in the sense of side-channel resistance of the code-based masking) that it has a larger dual distance, and also a lower kissing number for the same dual distance. Therefore, we introduce an ordering of different codes relying on their weight distributions as follows.

Definition 6 (Prefix-based lexicographical order of sequences). *Let (A_i) and (A'_i) $(0 \leq i \leq n)$ be two sequences of integers of length n. The sequence (A_i) is (strictly)* smaller *than the sequence (A'_i) if there exists $1 \leq j \leq n$, such that $A_i = A'_i$ for all $0 \leq i < j$, and $A_j < A'_j$.*

Definition 7 (Best weight distribution). *A linear code C is said to be* better *than a linear code C' if its weight distribution is (prefix-based)* smaller *than that of C'. A code has the* best *weight distribution if it is better than any other linear code.*

Thus, to obtain the best weight distribution, we apply the following three principles:

1. maximize the minimum distance d (recall that $d = \min\{i \neq 0, A_i > 0\}$)
2. (in case of a tie) minimize the kissing number A_d
3. (in case of a tie) minimize the following coefficients A_i, $i > d$ in lexicographical order.

Regarding the first principle, it is feasible to construct a maximum distance separable (MDS) code which maximizes the minimum distance. We have the following Delsarte's lemma for the dual of an MDS code.

Lemma 1 (Dual of an MDS code [10])**.** *The dual of an MDS code is also an MDS code.*

Corollary 1. *The dual distance of a linear MDS code of parameters $[n, k]_{2^\ell}$ is $d = k + 1$.*

Proof of the corollary. The dual distance of a linear MDS code is equal to the minimum distance of the dual of the code. which has parameters $[n, n-k]_{2^\ell}$. By Lemma 1, it is MDS. Therefore, the Singleton bound [22] is tight and we have that $n - (n-k) + 1 = d$. Hence $d = k + 1$. □

2.2 State-of-the-Art Results

Recall the communication channel-based setting of side-channel analysis [8,9] shown in Figure 1, with the following notations.

- $K, \hat{K}$ denote the secret and guessed key, respectively.
- T denotes the plaintext/ciphertext that can be accessed by an adversary.
- U is the sensitive variable which is encoded as V after code-based masking using an independent random mask M.
- The device leaks under leakage function f (typically Hamming weight $f = w_H$) so that $X = f(V)$.
- The side-channel leakage is modeled as $Y = X + N$ where typically $N \sim \mathcal{N}(0, \sigma^2)$ is an additive white Gaussian noise (AWGN).

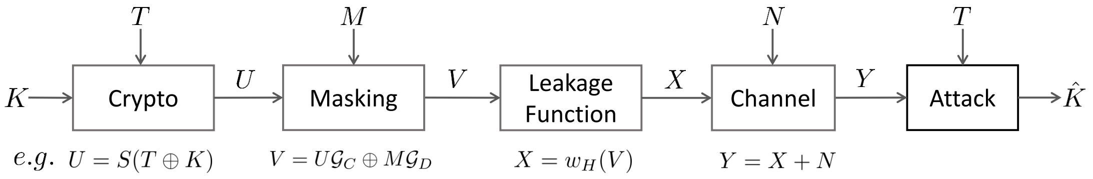


Figure 1: Side-channel leakage setup and subsequent analysis modelization (modified from [8]).

The Figure 1 makes use of the symbol "⊕" to denote finite field addition, and "+" for addition of reals. In the sequel, we focus on finite field operations: there is therefore no possible confusion. Hence we simply use "+" even in finite fields.

We consider the code-based masking of Figure 1 for which

$$V = U\mathcal{G}_C + M\mathcal{G}_D \tag{1}$$

where U and M are the sensitive variable and random mask, respectively. Two linear codes C and D with respective generator matrices $\mathcal{G}_C$ and $\mathcal{G}_D$ encode U and M into V.

It follows that from the perspective of side-channel resistance, the word-level security is only captured by the minimum distance of $D^\perp$ [5, 18]. By contrast, the bit-level security of a code-based masking is related to both the minimum distance and the kissing number of $D^\perp$ [6, 7] under the Hamming weight leakage model.

Rather than searching from all possible candidates as in [6], we aim at constructing optimal linear codes for GCM by an efficient algorithm. To the best of our knowledge, this is an open problem. It is known that a good code (for masking countermeasure) has a large minimum distance and a low kissing number [7]. However, we recall from Definition 4 that such kissing number is only one coefficient of the weight distribution polynomial. As we demonstrate in the sequel, the entire weight distribution is to be considered to assess the side-channel resistance of a code-based masking. As a consequence, we found that the best masking code for GCM is determined by Algorithm 1. In particular, the difference comparing with [6, 7] lies in line 4, which indicates the better code in case of a tie in A_i for $d \le i \le n$.

```
  Input  : Masking order $t$ (at word level over $\mathbb{F}_{2^\ell}$)
  Output : Codes for GCM over $\mathbb{F}_{2^\ell}$
1 Construct an MDS code $D$: $[n, n-k]_{2^\ell}$ with $d_D^\perp = t+1$   // Use Corollary 1, $d_D^\perp = n-k+1$
2 Apply subfield extension on $D$                                        // Use Definition 5
3 Compute the dual code $[D]^\perp$                                       // Use Definition 3
4 Choose the code $D$ such that $[D]^\perp$ has the best weight distribution   // Use Definition 7
5 return $D$
```

Algorithm 1: Finding the best masking code for GCM.

3 Orthogonal Bases and Subfield Representations

In a code-based masking scheme, the side-channel security order at bit level is related to the weight distribution of the codes in the subfield representation [6, 7]. Particularly, given a code D in (1) defined over $\mathbb{F}_{2^\ell}$, we wish to evaluate the weight distribution of the dual extended code $[D]^\perp$, and the natural question is to assess whether this is equivalent to evaluate the weight distribution of extended dual code $[D^\perp]$. However, as shown in Figure 2, the commutative relationship does not hold in general because depending on the choice of basis of $\mathbb{F}_{2^\ell}$ over $\mathbb{F}_2$, the two codes $[D]^\perp$ and $[D^\perp]$ are not always equivalent to each other.

$$\begin{tikzcd}
D \arrow[r, "Dual"] \arrow[d, "Subfield"'] & D^\perp \arrow[d, "Subfield"] \\
{[D]} \arrow[r, "Dual"'] & {[D]^\perp \overset{?}{=} [D^\perp]}
\end{tikzcd}$$

Figure 2: Commutative connection between sub-field representation and duality.

As it turns out, the commutative relationship will hold true if the basis used in subfield representation is a *trace-orthogonal* basis. Therefore, we first show how to construct trace-orthogonal bases and then investigate the subfield extension of the code.

3.1 Construction of Trace-Orthogonal Bases

Let $\ell > 1$ and $\mathbb{F}_{2^\ell}$ be the extension field of $\mathbb{F}_2$. By the Frobenius conjugacy property, the trace function $\mathrm{tr} : \mathbb{F}_{2^\ell} \to \mathbb{F}_2$, defined as $\mathrm{tr}(x) = \sum_{i=0}^{\ell-1} x^{2^i}$, is linear. The (trace-)orthogonality and orthonormality is defined as follows.

Definition 8. *Elements a_1, a_2 in $\mathbb{F}_{2^\ell}$ are* orthogonal *if $\mathrm{tr}(a_1 a_2) = 0$. A basis $\{a_1, a_2, \ldots, a_\ell\}$ of $\mathbb{F}_{2^\ell}$ over $\mathbb{F}_2$ is* orthonormal *if $\mathrm{tr}(a_i^2) = \mathrm{tr}(a_i) = 1$ and $\mathrm{tr}(a_i a_j) = 0$ for all $i \neq j$.*

Notice that, as mentioned in [21], we have the following result:

Lemma 2. *A (trace-)orthogonal basis in $\mathbb{F}_{2^\ell}$ is always orthonormal.*

Proof. Let a_i be elements in a basis, where $i \in \{1, \ldots, \ell\}$. We need to show that it satisfies $\mathrm{tr}(a_i) = 1$.

The trace takes values in $\mathbb{F}_2$, which consists in two elements, namely 0 and 1. Therefore, it must be proven that $\mathrm{tr}(a_i) \neq 0$. This means that a_i is not self-orthogonal, since $\mathrm{tr}(a_i^2) = \mathrm{tr}(a_i)^2 = \mathrm{tr}(a_i)$ in $\mathbb{F}_2$.

Let us reason by the absurd. Assume that a_i is self-orthogonal. Then, not only a_i is orthogonal to all vectors a_j ($j \neq i$), but also to itself. Therefore, it belongs to the dual of the space vector E generated by the basis $\{a_1, a_2, \ldots, a_\ell\}$. Notice that E is the universe code, hence its dual is the singleton $\{0\}$. Consequently $a_i = 0$, which contradicts the fact that a_i is a basis vector. □

Remark 1. *Incidentally, we notice that the condition (36) in [13, §5, p. 182] is superfluous, since already implied by condition (37).*

By [14, Note 3, page 75] (which points to the original paper [13]), we know that an orthonormal basis always exists. Although [13] gives a formal construction meant to provide the existence result, the resulting implementation is double-exponential in 2^ℓ, which is far too complex to implement in practice.

In this paper, we consider instead a fast, but probabilistic, trace-orthogonal basis construction given by Algorithm 2. For $\ell = 8$, it works most of the time in one iteration (e.g., about 70.20% over 2000 times of randomly running Algorithm 2). Examples are provided below.

Remark 2. *Strictly speaking, Algorithm 2 does not necessarily converge with a basis of full rank. We observed that depending on the scanning order of field elements at line 3, the algorithm can succeed or fail to return a basis. Therefore, we introduced a randomization at this line, and repeated the algorithm until it returns a (full rank) basis.*

In viewing of Definition 8, the elements in a basis must satisfy $\mathrm{tr}(a_i) \neq 0$. Therefore, we can improve Algorithm 2 by removing zero-trace elements with a preliminary check of all traces. The new procedure is shown in Algorithm 3.

Table 1 presents the comparison on efficiency between Algorithms 2 and 3. The performance metric is the execution time, measured on a server which runs the `Magma` system. This clearly shows the advantage of using Algorithm 3 when the order of the finite field is large. For instance, when $\ell = 16$, Algorithm 3 have a speedup by a factor of 5 compared to Algorithm 2.

We shall use the following two examples of trace-orthogonal bases throughout the rest of this paper:

```
Input  : $\ell \geq 1$, the extension order of $\mathbb{F}_2$, and $\alpha$, a primitive element of $\mathbb{F}_{2^\ell}$
Output : An orthonormal basis of $\mathbb{F}_{2^\ell}$
$(b_1, \ldots, b_\ell) \leftarrow (0, \ldots, 0)$                    // Basis, initialized with 0s
for $i \in \{1, \ldots, \ell\}$ do                                    // Find the $i$th element of the orthonormal basis
    for $a \in (\mathbb{F}_{2^\ell})^*$ do                            // Candidate next vector in the basis (chosen randomly)
        if $\mathrm{tr}(a) = 1$ then                                  // Test for $\mathrm{tr}(a^2) = \mathrm{tr}(a)^2 \neq 0$ (only element $\neq 0$ is 1 in $\mathbb{F}_2$)
            is_orthogonal ← true
            for $j \in \{1, \ldots, i-1\}$ do
                if $\mathrm{tr}(ab_j) \neq 0$ then                    // Test whether $a$ and $b_j$ are orthogonal
                    is_orthogonal ← false
            if is_orthogonal then
                $b_i \leftarrow a$
return $(b_1, \ldots, b_\ell)$
```

Algorithm 2: Randomized construction of an orthonormal basis in $\mathbb{F}_{2^\ell}$.

```
Input  : $\ell$, the extension order, and $\alpha$, a primitive element of $\mathbb{F}_{2^\ell}$
Output : An orthonormal basis of $\mathbb{F}_{2^\ell}$
list ← {}
for $i \in \{1, \ldots, 2^\ell - 1\}$ do                              // Check the trace of elements in $\mathbb{F}^*_{2^\ell}$
    if $\mathrm{tr}(\alpha^i) = 1$ then
        list ← list ∪ {i}                                             // Put the power in list if trace equals 1
$B \leftarrow \{\alpha^{\text{list}[1]}\}$                            // Create a set with one element
start ← 2                                                             // Set the start position of searching (can be changed)
while $\#B \neq \ell$ do
    $n$ ← start
    for $k \in \{2, \ldots, \ell\}$ do                                // Find the $k$th element of the orthonormal basis
        for $s \in \{n+1, \ldots, \#\text{list}\}$ do
            is_orthogonal ← true
            for $j \in \{1, \ldots, k-1\}$ do                         // Test whether the candidate is orthogonal with elements in B
                $a \leftarrow B[j] \cdot \alpha^{list[s]}$
                if $\mathrm{tr}(a) \neq 0$ then
                    is_orthogonal ← false
            if is_orthogonal then
                $B \leftarrow B \cup a$
                $n \leftarrow s$
        if $\#B < k$ then                                             // Start again if we cannot find next base
            break;
    start ← start + 1                                                 // Change a start position (if we do not get enough basis)
return $B$
```

Algorithm 3: The improved construction of orthonormal bases in $\mathbb{F}_{2^\ell}$.

- $\mathcal{B}_0 = \{\alpha^{252}, \alpha^{156}, \alpha^{122}, \alpha^{203}, \alpha^{5}, \alpha^{126}, \alpha^{71}, \alpha^{65}\}$,

Table 1: The comparison on efficiency of two algorithms for constructing trace-orthogonal bases. Note that with our `Magma` server is with Intel Xeon CPU@2.0GHz, 4 processors (only one is used), and with 16GB Memory.

ℓ		4	8	12	16	20	24
Run time (sec)	Alg. 2	0.0001	0.0038	0.1150	1.5034	36.0350	1146.1685
	Alg. 3	0.0001	0.0019	0.0334	0.3065	4.7267	267.7467

- $\mathcal{B}_1 = \{\alpha^{121}, \alpha^{252}, \alpha^{202}, \alpha^{20}, \alpha^{242}, \alpha^{15}, \alpha^{126}, \alpha^{44}\}$.

where α is the first primitive element in the finite field $\mathbb{F}_{2^8}$. Note that the irreducible polynomial used in this paper is: $g(\mathsf{X}) = \mathsf{X}^8 + \mathsf{X}^4 + \mathsf{X}^3 + \mathsf{X}^2 + 1$. Moreover, we also investigate the default basis used in `Magma`, which is a non-orthogonal basis:

- $\mathcal{B}_2 = \{1, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6, \alpha^7\}$.

3.2 Subfield Representation and Duality of Codes

We therefore specify the representation in Definition 5 by showing how to transform an element over $\mathbb{F}_{2^\ell}$ into $\mathbb{F}_2$. The subfield representation $[a]$ of a field element a is defined as follows.

Definition 9. *Let $b = (b_1, \ldots, b_\ell)$ an orthonormal basis of $\mathbb{F}_{2^\ell}$. The subfield representation of $a \in \mathbb{F}_{2^\ell}$ is $[a] = (\mathrm{tr}(ab_1), \ldots, \mathrm{tr}(ab_\ell))$.*

The subfield representation code $[D]$ can be seen a concatenated code (as per Forney [11]) with D of parameters $[n, k]_{2^\ell}$ as the outer code, and the universal $[\ell, \ell, 1]_2$ as the inner code. As a consequence, the side-channel security at bit level and word (ℓ-bit string) level are related by the subfield representation as follows: The security order at word level is the dual distance of the code in $\mathbb{F}_{2^\ell}$, whereas the security order at bit level is the dual distance of the subfield representation in $\mathbb{F}_2$.

A nice feature of trace-orthonormal bases is that duality and subfield representation commute:

Theorem 1. *Let D be a linear code. Then under a trace-orthogonal basis, we have:*

$$[D]^\perp = [D^\perp]. \tag{2}$$

Said equivalently, the duality and the sub-field representation form a commutative diagram:

$$\begin{tikzcd}
D \arrow[r, "Dual"] \arrow[d, "Subfield"'] & D^\perp \arrow[d, "Subfield"] \\
{[D]} \arrow[r, "Dual"] & {[D]^\perp = [D^\perp]}
\end{tikzcd}$$

Proof. Given $x, y \in \mathbb{F}_{2^\ell}^n$ and their subfield representations are $[x], [y] \in \mathbb{F}_2^{n\ell}$, respectively. Then the inner product $\langle x|y\rangle = 0$ implies that $0 = \mathrm{tr}(\langle x|y\rangle) = \sum_i \mathrm{tr}(x_i y_i) = \sum_i \sum_j [x_i]_j [y_i]_j = \langle [x] | [y] \rangle$ where the third equality holds because of the property of the trace-orthogonal basis. Therefore, we obtain $[D^\perp] \subseteq [D]^\perp$.

Inversely, two linear codes $[D^\perp]$ and $[D]^\perp$ are subspaces of $\mathbb{F}_2^{n\ell}$ that have the same length $2^{n\ell}$ and dimension $2^{(n-k)\ell}$, implying the same number of codewords in both codes. As a consequence, we have $[D^\perp] = [D]^\perp$. □

As a straightforward consequence of Theorem 1, the order of two transformations in lines 2 and 3 of Algorithm 1 are interchangeable. Therefore, the selection of the best codes can be achieved from the code D to the dual code $D^{\perp}$ and then to the subfield extension $[D^{\perp}]$.

Remark 3. *We notice that the resulting distances are not the same depending on:*

- *which basis is used,*
- *the code itself.*

We provide several examples of properties of codes $D^{\perp}$ of parameters $[5,3]_{256}$ (for $\ell = 8$). The `Magma` scripts are given in Appendix A). The difference between the tables are the bases:

- $\mathcal{B}_0$ is used in Table 2,
- $\mathcal{B}_1$ is used in Table 3.

Table 2: The dual distances for two seeds when drawing random code D, using $\mathcal{B}_0$ of $\mathbb{F}_{256}$.

`SetSeed(0)`		`SetSeed(1)`	
$d_{D^{\perp}}$	$d_{[D]^{\perp}}$	$d_{D^{\perp}}$	$d_{[D]^{\perp}}$
4	8	4	6
3	6	4	7
4	8	4	6
4	6	4	6
4	8	4	8
4	7	4	8
4	7	4	8
4	7	4	8
4	8	4	7
4	7	4	8

4 Characterizing Side-Channel Security by Weight Distribution

Mutual information (MI) is commonly used in tasks related to measuring side-channel leakage as an information-theoretic metric. Essentially, MI measures the statistical dependencies between the key-dependent variables and the leakage, which considers the full distributions of corresponding variables. Since the weight distribution determines how weights of codewords in a linear code are distributed, it therefore determines the leakage distribution of the masked variable from a coding-theoretic perspective [7].

In view of this, we have the following conjecture.

Conjecture 1. *MI between the sensitive variable and side-channel leakage depends on the weight distributions of the corresponding codes in the code-based masking.*

Table 3: The dual distances for two seeds when drawing random code D, using $\mathcal{B}_1$ of $\mathbb{F}_{256}$.

`SetSeed(0)`		`SetSeed(1)`	
$d_{D^\perp}$	$d_{[D]^\perp}$	$d_{D^\perp}$	$d_{[D]^\perp}$
4	8	4	7
3	6	4	7
4	7	4	7
4	7	4	8
4	8	4	7
4	7	4	7
4	7	4	8
4	6	4	7
4	7	4	7
4	7	4	8

It is well-known that for a code of dual distance d, any tuple of $d-1$ coordinates is uniformly distributed, and some tuples of d coordinates are linearly dependent [15, Theorem 10]. Therefore, the side-channel security order under probing model is $t = d - 1$, and an attack of order d, corresponding to codewords of Hamming weight equal to d, brings some mutual information that depends on σ^{-2d}, where σ^2 is the variance of the AWGN channel that characterized the leakage model. Moreover, since not all codewords have the same Hamming weight d, other codewords of weights $> d$ should bring more information when considering mutual information as an information-theoretic metric.

Said differently, the mutual information is related to $\sum_{i=0}^{n\ell} \frac{A_i}{\sigma^{2i}}$, or more precisely (removing the useless 1 constant arising from $i = 0$), it is related to:

$$\sum_{i=d}^{n\ell} \frac{A_i}{\sigma^{2i}},$$

where $n\ell$ is the length of the extended code over $\mathbb{F}_2$ and A_i is the number of codewords of weight i (in the dual of the code employed to mask the information). Hence the lexicographical order of the A_i to compare the amount of leakage is indeed associated with the masking code.

4.1 Connecting with Attacks

When evaluating with side-channel attacks, particularly in the optimal multivariate attacks (using higher-order optimal distinguishers) [4], the weight distribution also plays a significant role. More precisely, we have the following conjecture.

Conjecture 2. *The success rate of optimal multivariate attack is determined by the weight distributions of the corresponding codes in the code-based masking.*

Informally, as shown in Figure 1, given the same U, $w_H(V)$ is distributed as $w_H(V')$, where M and M' are uniformly drawn from two equivalent codes (because of the Hamming weight, which is coordinate-wise independent). Therefore, side-channel distinguishers should perform similarly when extracting key-dependent information from leakages under the Hamming weight model.

4.2 Numerical Results

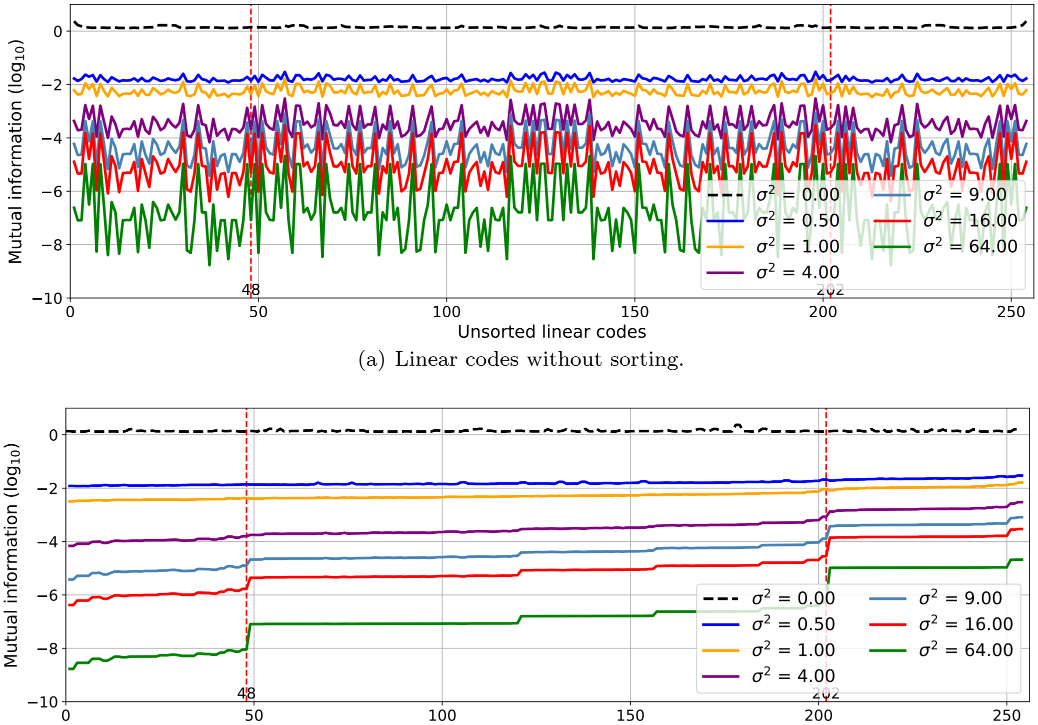
In the following, we consider a typical case of GCM by setting the generator matrices of the two codes C and D as follows:

$$\begin{aligned} \mathcal{G}_C & = \begin{pmatrix} 1 & 0 \end{pmatrix} , & (3)\\ \mathcal{G}_D = \begin{pmatrix} \alpha_1 & \alpha_2 \end{pmatrix} & = \begin{pmatrix} \alpha^i & \alpha^j \end{pmatrix} . & (4) \end{aligned}$$

Clearly, the code D is an MDS code of parameters $[2, 1, 2]$. Considering equivalent linear codes over $\mathbb{F}_{2^8}$, we can fix $\alpha^j = 1$ in $\mathcal{G}_D$. Hence there are only 254 candidates for the second element in $\mathcal{G}_D$, corresponding to 254 linear codes.

As a common setting in side-channel analysis, we take the Hamming weight leakage model with the Gaussian noise. The setup is shown in Figure 1 in a communication channel viewpoint. Considering different bases, we launch an information-theoretic evaluation on all linear codes under different noise levels. The results are shown in Figure 3, 4 and 5 for the three bases, respectively. In particular, we add Figure 3(a) for the purpose of comparison, which illustrates the effectiveness of our lexicographical order based sorting of all codes.

Note that the two vertical red dashed lines are for indicating the different dual distances $d_D^\perp \in \{2, 3, 4\}$. For instance in Figure 3(b), the first vertical line marked 48 means there are 48 linear codes with $d_D^\perp = 4$, and $202 - 48 = 154$ linear codes with $d_D^\perp = 3$, and remaining 52 linear codes with $d_D^\perp = 2$.



(a) Linear codes without sorting.

(b) Sorted linear codes in the lexicographical order.

Figure 3: Information-theoretic evaluation of all 254 candidates under the trace-orthogonal basis $\mathcal{B}_0$.

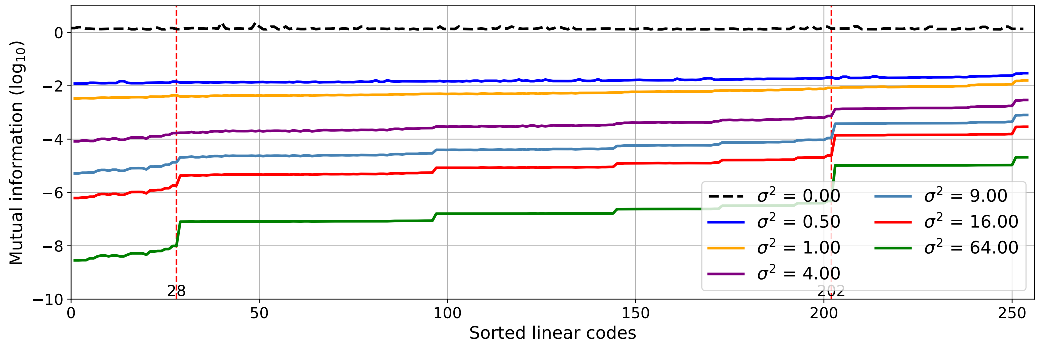


Figure 4: Information-theoretic evaluation of all 254 candidates under the trace-orthogonal basis $\mathcal{B}_1$ sorted in the lexicographical order.

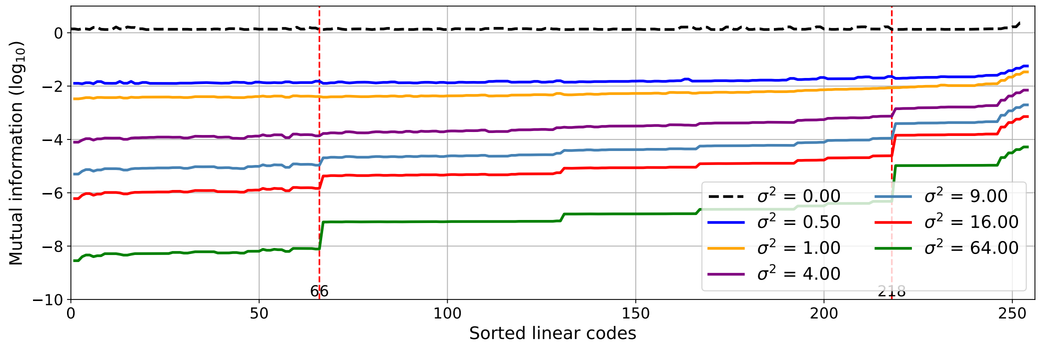


Figure 5: Information-theoretic evaluation of all 254 candidates under the default basis $\mathcal{B}_2$ sorted in the lexicographical order.

An interesting observation from Figure 3, 4 and 5 is, the bases have a significant impact on the distribution of linear codes. The mutual information increases (in most cases, except for some local minima) with the code lexicographic order on their weight enumeration polynomial. This justifies Conjecture 1. However, the number of exceptions (local minima) decreases when the noise increases, and the curves become indeed strictly increasing. Particularly, the first basis $\mathcal{B}_0$ gives the best weight distribution among the three bases, which will be investigated further in the next subsection.

4.3 Classifying Linear Codes

In order to find the best weight distributions under different bases, we classify linear codes as in Table 4. Specifically, in Table 4, we first show the distribution of the minimum distance of all 254 linear codes under the three bases, and then present the best weight distribution in the last column. The takeaway point for the three bases is that the basis has a significant impact on the distribution of the minimum distances. Under condition of the prefix-based lexicographical order of weight distribution (Definition 6), we focus on the number of codes with the minimum distance equal to 4, resulting that $\mathcal{B}_2$ gives more codes with $d = 4$ (among the three cases). On the contrary, the first basis $\mathcal{B}_0$ gives the best weight distribution among all three bases where $A_4 = 2$.

Secondly, we randomly generate 1,000,000 linear codes over $\mathbb{F}_2$ by fixing $n = 16$ and $k = 8$ for

Table 4: Classifying linear codes under different bases. Note that the float number in parenthesis is the ratio between the number of codes in a class and the total number of candidates.

	Subfield	Number of linear codes with different d					Best weight distribution
		$\#\{d=1\}$	$\#\{d=2\}$	$\#\{d=3\}$	$\#\{d=4\}$	$\#\{d=5\}$	
$\mathcal{B}_0$	$\mathbb{F}_{2^8} \to \mathbb{F}_2$	0	52 (0.2047)	154 (0.6063)	48 (0.1890)	0	[**1, 0, 0, 0, 2**, 22, 40, 44, 45, 40, 32, 20, 8, 2, 0, 0, 0]
$\mathcal{B}_1$	$\mathbb{F}_{2^8} \to \mathbb{F}_2$	0	52 (0.2047)	174 (0.6850)	28 (0.1102)	0	[**1, 0, 0, 0, 3**, 21, 38, 46, 45, 40, 34, 18, 7, 3, 0, 0, 0]
$\mathcal{B}_2$	$\mathbb{F}_{2^8} \to \mathbb{F}_2$	0	36 (0.1417)	152 (0.5984)	66 (0.2598)	0	[**1, 0, 0, 0, 4**, 22, 35, 42, 47, 46, 36, 14, 4, 4, 1, 0, 0]
Random codes	$\mathbb{F}_2$	60688 (0.0607)	357539 (0.3575)	528070 (0.5281)	53703 (0.0537)	0	[**1, 0, 0, 0, 1**, 23, 42, 42, 45, 40, 30, 22, 9, 1, 0, 0, 0]
BKLC	$\mathbb{F}_2$	0	0	0	0	1	[**1, 0, 0, 0, 0**, 24, 44, 40, 45, 40, 28, 24, 10, 0, 0, 0, 0]

comparison. The distribution of the minimum distances are listed in the fourth row of Table 4. One interesting observation is that this random approach gives a better weight distribution than all three bases over $\mathbb{F}_{2^8}$.

However, all above cases do not recover the best known linear code (BKLC) given $n = 16$ and $k = 8$. We know that there is a unique linear code with parameters $[16, 8, 5]$, which has the minimum distance equal to 5 [7]. Among all linear codes over $\mathbb{F}_2$, this BKLC code gives us the best weight distribution according to our lexicographical sorting, since it has $A_4 = 0$ while $A_4 > 0$ for other cases. From a perspective of side-channel analysis, this BKLC code provides us a masking code with the bit-level security order $t = d_D^\perp - 1 = 4$, that is higher than all other linear codes. Unfortunately, this code cannot be constructed by the subfield extension approach from $\mathbb{F}_{2^8}$ to $\mathbb{F}_2$ (e.g., by using bases like $\mathcal{B}_i$ for $i \in \{0, 1, 2\}$). This is also the reason why the direct sum masking can be better than the inner product masking in the sense of side-channel resistance [5, 7].

Evaluation of the best weight distributions under different bases. In Table 4, we present five best cases of the weight distribution. In order to have a fair comparison, we launch an information-theoretic evaluation by using mutual information. The results are depicted in Figure 6.

As shown in Figure 6, the main observation is that our lexicographical order-based sorting still works when comparing linear codes extended by using different bases. Note that for the best weight distribution under $\mathcal{B}_1$ and $\mathcal{B}_2$, the curve for $\mathcal{B}_1$ is slightly higher than that of $\mathcal{B}_2$. The reason is that other elements (e.g., A_{d+1}, A_{d+2}, etc) in the weight distribution under $\mathcal{B}_1$ have more impact on mutual information.

5 Discussion: Related Works

The problem of selecting optimal linear codes originates from [16] when choosing good codes for leakage squeezing (LS) scheme. It is latter considered in other schemes like low-entropy masking scheme (LEMS) [17] and direct sum masking (DSM) [3]. The problem also emerges in choosing good public parameters in IPM [1], since different parameters play a significant role in the side-channel resistance of IPM. Note that LS, IPM and DSM schemes are special cases

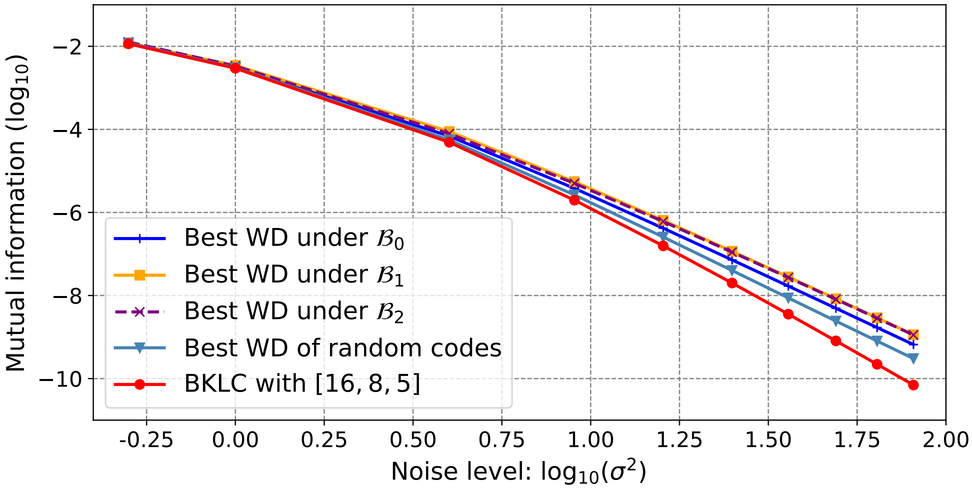


Figure 6: Information-theoretic evaluation of the best weight distributions (WD) under different bases as shown in Table 4.

of GCM as shown in [6]. Therefore, it is preferable to seek a solution to the problem in GCM as it is the most general case.

From the perspective of solution, using the dual distance as an indicator to choose good codes (in the sense of side-channel resistance) is proposed firstly in [3,5,17,18]. In particular, DSM and IPM are connected to each other over $\mathbb{F}_{2^\ell}$ and $\mathbb{F}_2$ in [5,18]. Then the kissing number proposed as the second indicator along with the dual distance is investigated in [6,7]. In viewing of the state-of-the-art results, this paper further extends the idea by using the full weight distribution and illustrate the exact conversion from $\mathbb{F}_{2^\ell}$ to $\mathbb{F}_2$ by giving the best weight distribution. In particular, we show how to use trace-orthogonal bases to obtain the extend codes over $\mathbb{F}_2$ irrespective to the order of two transformations, namely applying subfield representation first or computing dual codes first.

More generally, when the code-based masking is redundant [6], our approach also works in selecting optimal weight distribution. Considering the polynomial masking [19], which is based on Shamir's Secret Sharing (SSS) scheme, the kissing number should be replaced by the adjusted one (defined in [6], depending on both codes C and D in GCM). As a consequence, the selection of optimal linear codes should also use the adjusted weight distribution of C and D, rather than the weight distribution of D only in non-redundant cases like in IPM, etc.

6 Conclusions and Perspectives

In this work, we built a link between weight distribution of a linear code and the side-channel resistance of the corresponding code-based masking scheme. We first revisited the subfield extension of a linear code from word to bit level, which is related to word- and bit-level probing security. Using trace-orthonormal bases allowed us to have a commutative relationship of subfield representation and duality of a code. We then connected the side-channel resistance of the code-based masking to the weight distribution of corresponding linear codes. We have shown that the lexicographical ordering of the weight distribution can be used to find the best codes. More precisely, the lexicographic order on weight enumerators coincides with the information the corresponding codes leak as additive white Gaussian noise increases. Thus, the information-theoretic evaluation confirms the interest of the lexicographic sorting on weight distributions, which can be readily used to construct optimally resistant linear codes to side-channel attacks in our framework.

Acknowledgement

The authors would like to thank Patrick Solé, who suggested us to use the trace-orthogonal basis when building a connection between subfield representation and duality of the linear code, and also for insightful discussions.

A Magma Scripts

The Magma scripts used in Table 2 and 3 are as follows.

```
l := 8; // In this example, we consider the finite field GF(2, 8)
n := 5;
k := 3;
Nc := 10; // Obtain 10 random linear codes

SetSeed(0);
[{MinimumDistance(D), MinimumDistance(SubFieldRepresentationCode(D))}:
   D in [Dual(RandomLinearCode(GF(2,l),n,k)): i in {1..Nc}]];

SetSeed(1);
[{MinimumDistance(D), MinimumDistance(SubFieldRepresentationCode(D))}:
   D in [Dual(RandomLinearCode(GF(2,l),n,k)): i in {1..Nc}]];
```

Listing 1: Obtaining random linear codes, in Magma [23] language.

References

[1] Josep Balasch, Sebastian Faust, Benedikt Gierlichs, Clara Paglialonga, and François-Xavier Standaert. Consolidating Inner Product Masking. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 724–754. Springer, 2017.

[2] 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 *Advances in Cryptology - EUROCRYPT 2017, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 535–566, 2017.

[3] Julien Bringer, Claude Carlet, Hervé Chabanne, Sylvain Guilley, and Houssem Maghrebi. Orthogonal Direct Sum Masking - A Smartcard Friendly Computation Paradigm in a Code, with Builtin Protection against Side-Channel and Fault Attacks. In David Naccache and Damien Sauveron, editors, *Information Security Theory and Practice. Securing the Internet of Things - 8th IFIP WG 11.2 International Workshop, WISTP 2014, Heraklion, Crete, Greece, June 30 - July 2, 2014. Proceedings*, volume 8501 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2014.

[4] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks Will Fall Off – Higher-Order Optimal Distinguishers. In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology – ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*, volume 8874 of *Lecture Notes in Computer Science*, pages 344–365. Springer, 2014.

[5] Claude Carlet and Sylvain Guilley. Statistical properties of side-channel and fault injection attacks using coding theory. *Cryptography and Communications*, 10(5):909–933, 2018.

[6] Wei Cheng, Sylvain Guilley, Claude Carlet, Jean-Luc Danger, and Sihem Mesnager. Information leakages in code-based masking: A unified quantification approach. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(3):465–495, 2021.

[7] Wei Cheng, Sylvain Guilley, Claude Carlet, Sihem Mesnager, and Jean-Luc Danger. Optimizing Inner Product Masking Scheme by a Coding Theory Approach. *IEEE Trans. Inf. Forensics Secur.*, 16:220–235, 2021.

[8] Wei Cheng, Yi Liu, Sylvain Guilley, and Olivier Rioul. Attacking masked cryptographic implementations: Information-theoretic bounds. *CoRR*, abs/2105.07436, 2021.

[9] Éloi 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.

[10] Philippe Delsarte. The association schemes of coding theory. In *Combinatorics*, pages 143–161. Springer, Dordrecht, 1975. DOI: `10.1007/978-94-010-1826-5_7`.

[11] G. David Forney. *Concatenated codes.* PhD thesis, M.I.T. Dept. of Electrical Engineering, December 1965.

[12] Yuval Ishai, Amit Sahai, and David Wagner. Private Circuits: Securing Hardware against Probing Attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, August 17–21 2003. Santa Barbara, California, USA.

[13] Abraham Lempel. Matrix Factorization Over GF(2) and Trace-Orthogonal Bases of GF(2^{n}). *SIAM J. Comput.*, 4(2):175–186, 1975.

[14] Rudolf Lidl and Harald Niederreiter. Encyclopedia of Mathematics and Its Applications #20. Cambridge University Press, 1997. ISBN 10: 0521392314, ISBN 13: 9780521392310.

[15] F. Jessie MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes.* Elsevier, Amsterdam, North Holland, 1977. ISBN: 978-0-444-85193-2.

[16] Houssem Maghrebi, Sylvain Guilley, and Jean-Luc Danger. Leakage squeezing countermeasure against high-order attacks. In Claudio Agostino Ardagna and Jianying Zhou, editors, *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication - 5th IFIP WG 11.2 International Workshop, WISTP 2011, Heraklion, Crete, Greece, June 1-3, 2011. Proceedings*, volume 6633 of *Lecture Notes in Computer Science*, pages 208–223. Springer, 2011.

[17] Maxime Nassar, Youssef Souissi, Sylvain Guilley, and Jean-Luc Danger. RSM: A small and fast countermeasure for AES, secure against 1st and 2nd-order zero-offset SCAs. In Wolfgang Rosenstiel and Lothar Thiele, editors, *2012 Design, Automation & Test in Europe Conference & Exhibition, DATE 2012, Dresden, Germany, March 12-16, 2012*, pages 1173–1178. IEEE, 2012.

[18] Romain Poussier, Qian Guo, François-Xavier Standaert, Claude Carlet, and Sylvain Guilley. Connecting and Improving Direct Sum Masking and Inner Product Masking. In Thomas Eisenbarth and Yannick Teglia, editors, *Smart Card Research and Advanced Applications - 16th International Conference, CARDIS 2017, Lugano, Switzerland, November 13-15, 2017, Revised Selected Papers*, volume 10728 of *Lecture Notes in Computer Science*, pages 123–141. Springer, 2017.

[19] Emmanuel Prouff and Thomas Roche. Higher-Order Glitches Free Implementation of the AES Using Secure Multi-party Computation Protocols. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES*, volume 6917 of *LNCS*, pages 63–78. Springer, 2011.

[20] 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.

[21] Gadiel Seroussi and Abraham Lempel. Factorization of Symmetric Matrices and Trace-Orthogonal Bases in Finite Fields. *SIAM J. Comput.*, 9(4):758–767, 1980.

[22] Richard C. Singleton. Maximum distance q-nary codes. *IEEE Trans. Information Theory*,

10(2):116–118, 1964. DOI: 10.1109/TIT.1964.1053661.

[23] University of Sydney (Australia). Magma Computational Algebra System. http://magma.maths.usyd.edu.au/magma/, Accessed on 2021-08-22.

[24] Weijia Wang, Pierrick Méaux, Gaëtan Cassiers, and François-Xavier Standaert. Efficient and Private Computations with Code-Based Masking. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(2):128–171, 2020.