HAL
open science

Toward finding best linear codes for side-channel protections (extended version)

Wei Cheng, Yi Liu, Sylvain Guilley, Olivier Rioul

To cite this version:

Wei Cheng, Yi Liu, Sylvain Guilley, Olivier Rioul. Toward finding best linear codes for side-channel protections (extended version). Journal of Cryptographic Engineering, 2022, 10.1007/s13389-022-00305-x . hal-04136993

HAL Id: hal-04136993

https://telecom-paris.hal.science/hal-04136993v1

Submitted on 6 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards Finding Best Linear Codes for Side-Channel Protections (Extended Version*)

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

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, Palaiseau, 91120, France.
Email: firstname.lastname@telecom-paris.fr.
[2]Secure-IC S.A.S., 104 Boulevard du Montparnasse, Paris, 75014, France
Email: firstname.lastname@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 empowered attacks.

It is thus extremely important to ensure some reliable protection against those attacks. Countermeasures are optimized accordingly, favoring those whose implementation has mathematically provable security. For this reason, random masking [17, 28] has turned out to be the countermeasure of reference.

Recently, the *generalized code-based masking* (GCM) [8, 33] has been promoted as a coding-theoretic way to unite several masking schemes. The peculiarities of inner product masking, direct

* This work is an extended version of [12], which has been published in PROOFS'21.

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, the corresponding linear codes in GCM are naturally built with $\mathbb{F}_{2^\ell}$ as the base field.

However, optimizing the linear 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 [17] which only considered bits. Current probing models encompass probing of full-width registers [28].
- 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 canceled out completely, the effect of the masking. 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 [7, 24]. Also, the bit-level security relates to the extension of the code into the base field [7, 9]. 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 (TOBs), 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 [21]) 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 [21] 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 [21] 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 [21]) 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 $\mathbb{F}_2$. For a vector $c \in \mathbb{F}_{2^\ell}^n$, we shall note $[c]$ the broadcast extension of every component, meaning $[c] = ([c_1], \ldots, [c_n])$.

The subfield extension $[C]$ is the set of all vectors obtained from the codewords of C by taking the subfield representation, i.e., $[C] = ([c], c \in C)$.

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 [8, 9], 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, that integrates both the minimum distance and the kissing numbers.

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 with the same code parameters n and k.

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 [15]) 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 [31] is tight and we have that $n-(n-k)+1 = d$. Hence $d = k + 1$. □

We finally introduce the optimal linear codes over $\mathbb{F}_{2^\ell}$ given parameters n and k as follows.

Definition 8 ((d, A_d)-Optimal linear code) A linear code C of parameter $[n, k]$ over $\mathbb{F}_{2^\ell}$ is said to be (d, A_d)-optimal if its subfield extension $[C]$ has the largest minimum distance d and the lowest kissing number A_d.

The important case is that of a (d, A_d)-optimal binary linear code over the binary field $\mathbb{F}_2$. For instance, two optimal binary linear codes are: $[8, 4, 4]_2$, that is $(4, 14)$-optimal, and $[16, 8, 5]_2$ that is $(5, 24)$-optimal, respectively. While there are constructions of MDS codes over $\mathbb{F}_{2^\ell}$ under condition $n < 2^\ell$, the determination of (d, A_d)-optimal linear codes is still an open problem except for

trivial cases like repetition codes or parity check codes (both of them being trivial MDS codes).

In this paper, we focus on the *binary* extensions over $\mathbb{F}_2$ for two reasons. First, the side-channel leakage originates from bits (e.g., wires, registers, memory elements, etc) of running devices. Secondly, as demonstrated by information-theoretic evaluations [9] and attack-based evaluations [10], the two ingredients of (d, A_d)-optimal linear codes indeed indicate the side-channel resistance of code-based protections.

Remark 1 A linear code with the best weight distribution is also a (d, A_d)-optimal code, but the converse is not always true in the sense that not all binary linear codes can be mapped into a linear code over $\mathbb{F}_{2^\ell}$.

Remark 2 The kissing number in Definition 8 should be replaced by the adjusted kissing number [8] when the two linear codes C and D in code-based masking are not complementary.

2.2 State-of-the-Art Results

Recall the communication channel-based setting of side-channel analysis [11, 14] 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 leakage model $f = w_H$) so that $X = f([V])$, where $[V]$ denotes element-wise subfield representation when V is a vector.
- 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). In addition, N shall be a multivariate Gaussian variable, e.g., in the presence of masking, when V is a vector.

The Figure 1 makes use of the symbol "$\oplus$" 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 $D^\perp$ [7, 24]. By contrast, the bit-level security of a code-based masking is related to both the minimum distance and the kissing number of $D^\perp$ [8, 9] under the Hamming weight leakage model.

Rather than searching from all possible candidates as in [8], 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 [9]. 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 [8, 9] lies in line 4, which indicates the better code in case of a tie in A_i for $d \le i \le n$.

Algorithm 1: Conceptual process for finding the best masking code for GCM.

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 to get $[D]$ // Use Def. 5
3 Compute the dual code $[D]^\perp$ // Use Def. 3
4 **if** $[D]^\perp$ *has the best weight distribution* **then** // Use Def. 7
5 | **return** D
6 **else**
7 | **goto** Line 1
8 **end**

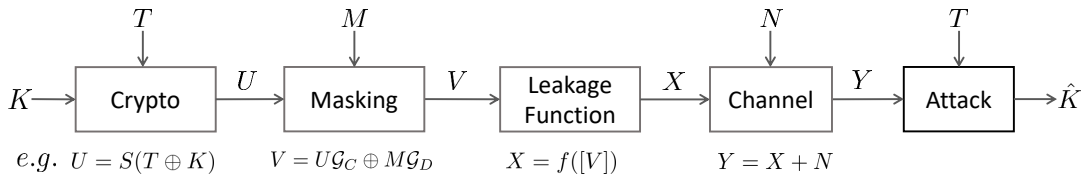


Fig. 1 Side-channel leakage setup and subsequent analysis modelization (modified from [11]).

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 [8, 9]. 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.

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

Fig. 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 9 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 [30], 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 $\mathbb{F}_2$.

Assume on the contrary 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 $\{a_1, a_2, \ldots, a_\ell\}$ (the universe code), whose dual is the singleton $\{0\}$. Consequently $a_i = 0$, which contradicts the fact that a_i is a basis vector. □

Remark 3 Incidentally, we notice that the condition (36) in [19, Chap 5, p. 182] is superfluous, since already implied by condition (37).

By [20, Note 3, page 75] (which points to the original paper [19]), we know that an orthonormal basis always exists. Although [19] 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 4 Strictly speaking, Algorithm 2 does not necessarily converge with a basis of full rank. We observed

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

Input : $\ell \geq 1$, the extension order of $\mathbb{F}_2$
Output : An orthonormal basis of $\mathbb{F}_{2^\ell}$

1 $(b_1, \ldots, b_\ell) \leftarrow (0, \ldots, 0)$ // Basis, initialized with 0s
2 **for** $i \in \{1, \ldots, \ell\}$ **do** // Find the ith element of the orthonormal basis
3 **for** $a \in (\mathbb{F}_{2^\ell})^*$ **do** // Candidate next vector in the basis (chosen randomly)
4 **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 $\mathbb{F}_2$)
5 is_orthogonal $\leftarrow$ true
6 **for** $j \in \{1, \ldots, i-1\}$ **do**
7 **if** $\mathrm{tr}(ab_j) \neq 0$ **then** // Test whether a and b_j are orthogonal
8 is_orthogonal $\leftarrow$ false
9 **end**
10 **end**
11 **if** is_orthogonal **then**
12 $b_i \leftarrow a$
13 **end**
14 **end**
15 **end**
16 **end**
17 **return** $(b_1, \ldots, b_\ell)$

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 9, 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:

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

Input : ℓ, the extension order, and α, a primitive element of $\mathbb{F}_{2^\ell}$
Output : An orthonormal basis of $\mathbb{F}_{2^\ell}$

1 list $\leftarrow \{\}$ // Create an empty list
2 **for** $i \in \{1, \ldots, 2^\ell - 1\}$ **do** // Check the trace of elements in $\mathbb{F}^*_{2^\ell}$
3 **if** $\mathrm{tr}(\alpha^i) = 1$ **then**
4 list $\leftarrow$ list $\cup$ {i} // Put the power in list if trace equals 1
5 **end**
6 **end**
7 $B \leftarrow \{\alpha^{\mathsf{list}[1]}\}$ // Create a set with one element
8 start $\leftarrow 2$ // Set the start position of searching (can be changed)
9 **while** $\#B \neq \ell$ **do**
10 $n \leftarrow$ start //
11 **for** $k \in \{2, \ldots, \ell\}$ **do** // Find the kth element of the orthonormal basis
12 **for** $s \in \{n+1, \ldots, \#\mathsf{list}\}$ **do**
13 is_orthogonal $\leftarrow$ true
14 **for** $j \in \{1, \ldots, k-1\}$ **do** // Test whether the candidate is orthogonal with elements in B
15 $a \leftarrow B[j] \cdot \alpha^{list[s]}$
16 **if** $\mathrm{tr}(a) \neq 0$ **then**
17 is_orthogonal $\leftarrow$ false
18 **end**
19 **end**
20 **if** is_orthogonal **then**
21 $B \leftarrow B \cup a$
22 $n \leftarrow s$
23 **end**
24 **end**
25 **if** $\#B < k$ **then** // Start again if we cannot find next base
26 **break**
27 **end**
28 **end**
29 start $\leftarrow$ start $+ 1$ // Change a start position (if we do not get enough basis)
30 **end**
31 **return** B

- $\mathcal{B}_0 = \{\alpha^{252}, \alpha^{156}, \alpha^{122}, \alpha^{203}, \alpha^{5}, \alpha^{126}, \alpha^{71}, \alpha^{65}\}$,
- $\mathcal{B}_1 = \{\alpha^{121}, \alpha^{252}, \alpha^{202}, \alpha^{20}, \alpha^{242}, \alpha^{15}, \alpha^{126}, \alpha^{44}\}$.

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: $g(\mathsf{X}) = \mathsf{X}^8 + \mathsf{X}^4 + \mathsf{X}^3 + \mathsf{X}^2 + 1$.

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

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 10 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 [16]) 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 is 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 $[D^\perp]$. Section 3.3 illustrates the gain in terms of speed of this method.

Remark 5 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.

Therefore, the main takeaway point is that the bases have significant impact on the coding-theoretic properties of the extended codes.

3.3 Optimized Searching Method

We notice that the Subfield extension operation is "one-way". Namely, it is easy to extend a code from $\mathbb{F}_{2^\ell}$ to $\mathbb{F}_2^\ell$ (see Magma

Table 2 The dual distances for two seeds when drawing 10 random codes 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

Table 3 The dual distances for two seeds when drawing 10 random codes 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

`SubfieldRepresentationCode` command), but the inverse operation is not trivial. Moreover, not all codes of parameters $[n\ell, k\ell]_2$ can be interpreted as codes $[n, k]_{2^\ell}$. On the contrary, taking the dual of a linear code is invertible, and even involutive, as $(C^\perp)^\perp = C$.

Thus, leveraging trace-orthogonal bases, one can simplify the search for good codes by trading Alg. 4 (which is a realization of Alg. 1) by Alg. 5, in particular, saving the computation of the dual codes.

Algorithm 4: Bounded search for an efficient code

Input : Number of iterations N
Output : Best found GCM code over $\mathbb{F}_{2^\ell}$

```
w ← (2^n, 0, ..., 0)   // Worst case for a weight enumeration polynomial
D_best ← RandomCode [n, k]_{2^ℓ}
for i ∈ {1, ..., N} do
    Select a random code D
    if enumerationPolynomial ([D]^⊥) is better than w then
        w ← enumerationPolynomial ([D]^⊥)
        D_best ← D
    end
end
return D_best
```

Algorithm 5: Optimized (compared to Alg. 4) bounded search for an efficient code

Input : Number of iterations N
Output : Best found GCM code over $\mathbb{F}_{2^\ell}$

```
w ← (2^n, 0, ..., 0)   // Worst case for a weight enumeration polynomial
D_best ← RandomCode[n, k]_{2^ℓ}
for i ∈ {1, ..., N} do
    Select a random code D'
    if enumerationPolynomial ([D']) is better than w then   // No computation of dual code for all candidates
        w ← enumerationPolynomial ([D'])
        D_best ← D^⊥   // This operation has been procrastinated
    end
end
return D_best
```

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.

Consider a linear leakage model including the Hamming weight model. 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 [9].

In view of the above reasoning, we have the following conjecture.

Conjecture 1 MI between the sensitive variable and side-channel leakage under linear leakage models (e.g., Hamming weight model or weighted sum of bits model) depends on the weight distributions of the corresponding codes in the code-based masking [*].

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 [21, 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 [9]. 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, as inspired by [9, Theorem 4][†], 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:

$$\sum_{i=d}^{n\ell} \frac{A_i}{\sigma^{2i}}, \tag{3}$$

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.

[*]It is worth noting that, in theory, MI is not restricted to rely on specific assumption on the leakage model. However, we focus on the linear leakage model in this paper.

[†]Note that Theorem 4 in [9] only focuses on the first A_i for non-zero codewords.

4.1 Illustrating the Impact of Weight Distributions

An illustration of the terms A_i/σ^{2i} for $0 \leq i \leq 16$ (recall (3)), in the case two masking schemes corresponding to $\ell = 8$ and $n = 2$, is provided in Fig. 3. The two represented codes are:

- Boolean masking, and
- Masking using the first code in Tab. 4.

The values which are represented are A_i/σ^{2i}; the value for which i is equal to the minimum distance is an approximation of the mutual information. Such dominating coefficient is shown in Fig. 3 with larger symbol $\times$ or $+$. The figure shows the values in logarithmic scale; the null values are not represented (for the Boolean masking case, weight coefficient is equal to zero for even values of i). It is recalled from [9] how the code impacts the security. Consider the largest symbol:

- its *abscissa* corresponds to the leakage order (e.g., with different weights), whereas
- its *ordinate* corresponds to the leakage amplitude (approximate information leakage) for that leakage order.

It is clear from Fig. 3 that code-based masking performs better than Boolean masking on both metrics.

The impact of non-linear masking, such as (6) below, is to introduce combination(s) of bits before the attacker tries himself to further combine bits in his high-order attack. Therefore, the attacker needs to combine less bits (since some are already combined) to perform a successful attack.

Now, the Leakage Function box in Fig. 1 may consist in two physical phenomena:

1. Upfront (at its inputs): the design features *cross-talks* for instance, bits are combined. The combination is often non-linear (e.g., the concrete example discussed in Sec. 4.4), in that for instance the leakage of one bit is strengthened when another (nearby) bit is having a given value or experiencing a given transition. Hence a non-linear leakage model, even before any "noisy leakage" has occurred.
2. Downstream (at its outputs): the side-channel antenna is large, some aggregation (under the form of a linear combination) is performed and turns Boolean values into a real number.

The *probing model* operates at the input of the Leakage Function box, whereas the *bounded moments leakage model* operates at the output of the Channel box. Both models should be considered simultaneously in evaluating practical security of cryptographic implementations, since the attacker has the choice of its weapon.

4.2 Connecting with Attacks

When evaluating with side-channel attacks, particularly in the optimal multivariate attacks (using higher-order optimal distinguishers) [6], the weight distribution also plays a significant role. More precisely, we have the following conjecture.

Conjecture 2 The success rate of optimal multivariate attack under linear leakage models 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.

4.3 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:

$$\mathcal{G}_C = \begin{pmatrix} 1 & 0 \end{pmatrix}, \tag{4}$$

$$\mathcal{G}_D = \begin{pmatrix} \alpha_1 & \alpha_2 \end{pmatrix} = \begin{pmatrix} \alpha^i & \alpha^j \end{pmatrix}. \tag{5}$$

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 4, 5 and 6 for the three bases, respectively. In particular, we add Figure 4(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 4(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 $d_D^{\perp} = 2$.

An interesting observation from Figure 4, 5 and 6 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.

4.4 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 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 $\mathbb{F}_{2^8}$.

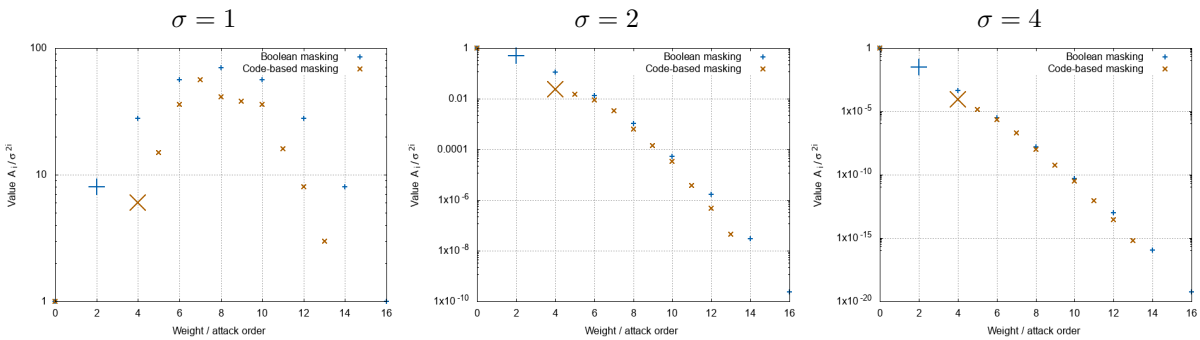
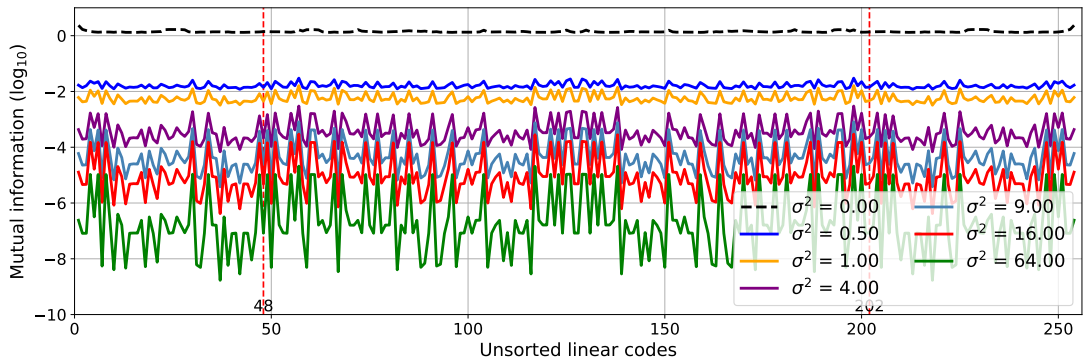
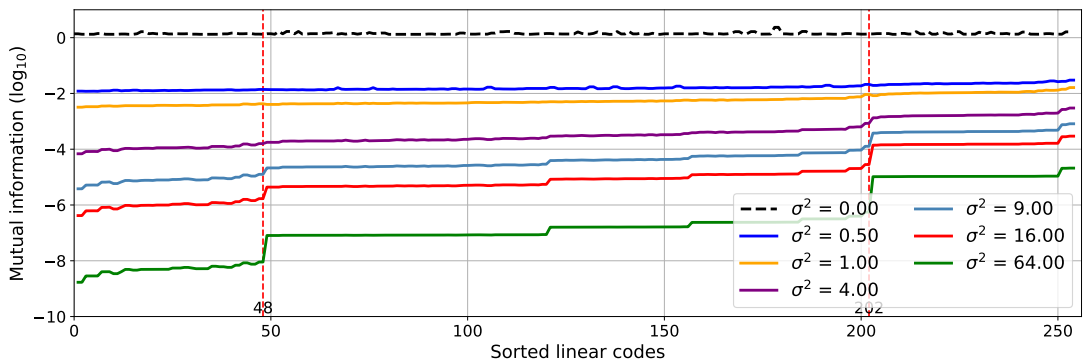


Fig. 3 Value of A_i/σ^{2i} per weight (or equivalently, per attack order). Largest symbol indicates leading term.



(a) Linear codes without sorting.



(b) Sorted linear codes in the lexicographical order.

Fig. 4 Information-theoretic evaluation of all 254 candidates under the trace-orthogonal basis $\mathcal{B}_0$.

However, all above cases do not recover the best known linear code (referred to as BKLC in Magma) 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 [9]. 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 [7, 9].

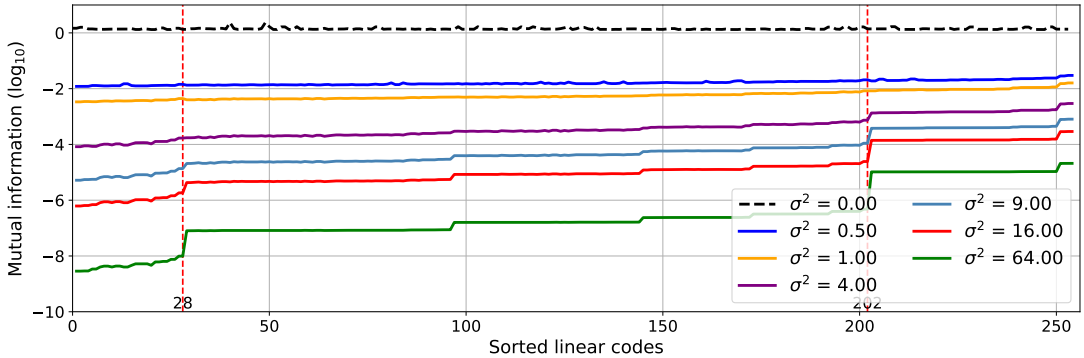


Fig. 5 Information-theoretic evaluation of all 254 candidates under the trace-orthogonal basis $\mathcal{B}_1$ sorted in the lexicographical order.

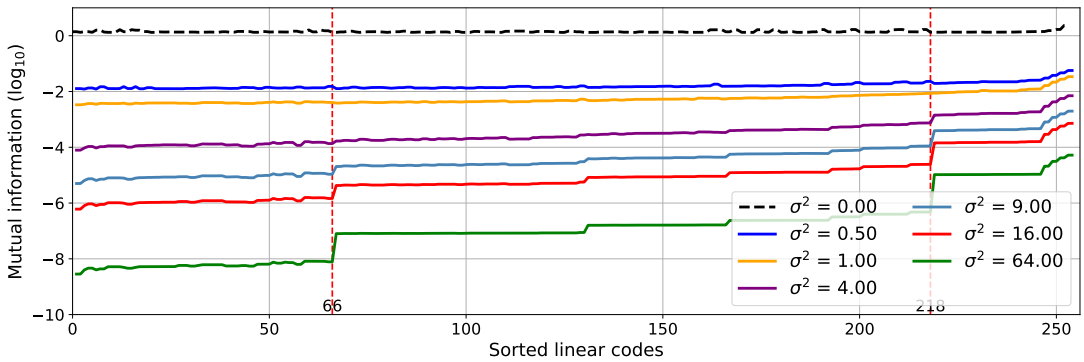


Fig. 6 Information-theoretic evaluation of all 254 candidates under the default basis $\mathcal{B}_2$ sorted in the lexicographical order.

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	Optimal codes
		$\#\{d=1\}$	$\#\{d=2\}$	$\#\{d=3\}$	$\#\{d=4\}$	$\#\{d=5\}$		
$\mathcal{B}_0$	$\mathbb{F}_{2^8} \to \mathbb{F}_2$	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]	$(4,2)$-optimal
$\mathcal{B}_1$	$\mathbb{F}_{2^8} \to \mathbb{F}_2$	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]	$(4,3)$-optimal
$\mathcal{B}_2$	$\mathbb{F}_{2^8} \to \mathbb{F}_2$	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]	$(4,4)$-optimal
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]	$(4,1)$-optimal
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]	$(5,24)$-optimal

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 7.

As shown in Figure 7, 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.

The generator matrices of the optimal codes in Table 4 are listed in Appendix B.

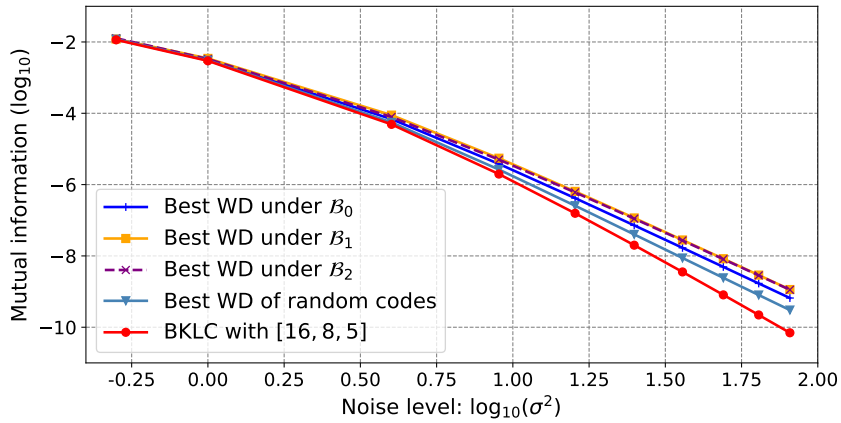


Fig. 7 Information-theoretic evaluation of the best weight distributions (WD) under different bases as shown in Table 4.

4.5 On Another Leakage Model

Although we mainly focus on the linear leakage models in this work, our analysis on using good linear codes can also be applied to other leakage models like the Hamming distance (HD) model. Let $V = (V_1, V_2)$ be the two shares in 2-share masking. Consider that V_1 and V_2 are manipulated consecutively, e.g., loaded in one register, we shall have the following HD leakage:

$$\begin{aligned} d_H(V_1,V_2) &= w_H(V_1 \oplus V_2) \\ &= w_H(V_1) + w_H(V_2) - 2w_H(V_1 \wedge V_2) \end{aligned} \tag{6}$$

where d_H is the Hamming distance function, and $\wedge$ is the bit-wise AND operator.

Therefore, the HD leakage as in (6) is a kind of second-order leakages, resulting in a decreased (effective) security order of the corresponding masking. For instance, a first-order (2-share) Boolean masking can be compromised under this HD leakage because the dual distance of the corresponding linear codes is only 2.

Fortunately, the code-based masking with the best linear codes can still work. Specifically, taking those linear codes with the dual distance equal to 3 or 4 in Table 4 can very well resist to the above HD leakage since HD leakage is a kind of second-order leakages [9]. Another similar scenario can happen with the transitional leakage caused by physical defaults like couplings [1, 13]. More general leakage models shall also be included and characterized as in [9, Theorem 1].

From an evaluation perspective, practical attack-based investigations in [35] demonstrate the advantages of utilizing the best linear codes, against both the template attack and higher-order correlation attacks. Moreover, this work also shows that code-based masking with good linear codes can resist transitional leakages in practice. To summarize, even with different leakage models, our approach can provide better choices in enhancing the side-channel resistance of masked cryptographic implementations.

5 Discussion: Related Works

In this section, we first review the selection of optimal linear codes in code-based masking and then give some examples of optimal codes in the literature, particularly with one upper bound on the minimum distance of the extend binary linear codes.

5.1 Linear Codes in Code-based Masking

The problem of selecting optimal linear codes originates from [22] when choosing good codes for leakage squeezing (LS) scheme. It is latter considered in other schemes like low-entropy masking scheme (LEMS) [23] and direct sum masking (DSM) [5]. 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 of GCM as shown in [8]. 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 [5, 7, 23, 24]. In particular, DSM and IPM are connected to each other over $\mathbb{F}_{2^\ell}$ and $\mathbb{F}_2$ in [7, 24]. Then the kissing number proposed as the second indicator along with the dual distance is investigated in [8, 9]. 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 [8], our approach also works in selecting optimal weight distribution. Considering the polynomial masking [25], which is based on Shamir's Secret Sharing (SSS) scheme, the kissing number should be replaced by the adjusted one (defined in [8], 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.

5.2 An Upper Bound on the Minimum Distance

The investigation of coding-theoretic properties of the extended linear codes over the subfield has been the topic of several works [3, 4, 18, 26, 27, 29, 34]. The subfield extension of a linear code over $\mathbb{F}_2$ is usually called its binary image. As we show in this paper, the coding-theoretic properties (e.g., the minimum distance, weight distribution, etc) usually depend on both the code itself and the bases that used for subfield extension [4, 26, 27], while under certain conditions [3], the minimum Hamming weight shall be independent of the bases.

As our aim is the selection of optimal linear codes, one natural question that raises is how to (upper) bound the minimum distance of the binary images. Interestingly, Rabizzoni [26, Theorem 1] propose both the upper and lower bounds in this respect (applied to $\mathbb{F}_{2^\ell}$):

$$d \leq d' \leq \left\lceil \frac{d \cdot \ell \cdot 2^{\ell-1}}{2^\ell - 1} \right\rceil, \tag{7}$$

where d and d' denote the minimum distances of the linear code and its binary image, $\lceil x \rceil$ is the greatest integer less than or equal to x. In particular, *r.h.s* of (7) is smaller than $d \cdot \ell$ when $\ell > 1$, where the latter is a trivial upper bound on d'.

In Table 5, we present several examples of the optimal linear codes with $\ell = \{4, 8\}$ for $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^8}$, respectively. Note that d_{BKLC} denotes the minimum distances of the best known linear codes over $\mathbb{F}_2$ given by Magma database; d_{up} is for the upper bound from *r.h.s* of (7). From Table 5, one can observe that the upper bound (by *r.h.s*) will be looser when ℓ gets larger, while interestingly, when $\ell = 4$, the upper bound is very close, or even exact when $n = 3$ and $k = 2$.

Table 5 Validation of the upper bound on the minimum distance of binary images. The detailed constructions of those codes can be found in [9, 10].

Code Parameters	$\mathbb{F}_{2^\ell}$	d	d'	d_{BKLC}	d_{up} by (7)
$n = 2,\ k = 1$	$\ell = 4$	2	3	**4**	**4**
	$\ell = 8$	2	4	5	8
$n = 3,\ k = 2$	$\ell = 4$	3	**6**	**6**	**6**
	$\ell = 8$	3	8	8	12

However, the upper bound by (7) is only related to d and the degree of the finite field $\mathbb{F}_{2^\ell}$, while it is independent of code parameters n and k. Intuitively, it shall be tighter by considering these parameters. We leave this problem open for further investigations.

5.3 Impacts of Linear Codes on Efficiency

One main disadvantage of code-based masking is its higher overhead compared with the simplest Boolean masking since more operations like finite field multiplication are involved in code-based masking. Taking real-world implementation-based evaluations on an embedded AVR microcontroller [1] and an ARM Cortex M4 board [35], the number of clock cycles of IPM is about 1.2 to 1.5 times to the Boolean counterpart with the same number of shares.

However, the different choices of linear codes in IPM have no significant impact on the efficiency, except the trivial case of Boolean masking is considered. For instance, setting the four linear codes in IPM [35] leads to the same clock cycle counts since this software implementation is designed to be constant time. Similarly, taking the best linear codes as we proposed in this work should have no significant impact on software-based implementations. Still, the hardware-based implementation shall be improved by carrying out operations over $\mathbb{F}_2$, and we leave this problem open for future investigation.

6 Conclusion and Perspective

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. As a perspective, we intend to consider practical applications in designing efficient masked cryptographic implementations of high-order security.

Acknowledgments. 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.

This work has partially benefited from the bilateral MESRI-BMBF project "APRIORI" from the ANR cybersecurity 2020 call. It is also part of the Horizon 2020 "SPARTA" project under grant agreement number 830892. Besides, the authors acknowledge financial support of the French national Bank (BPI) under Securyzr-V grant (Contract n° DOS0144216/00), a RISC-V centric platform integrating security co-processors.

Appendix A Magma Scripts

The Magma script used to generate results in Table 2 and 3 is given in Listing 1.

Listing 1 Obtaining random linear codes, in Magma [32] language.

```
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}]];
```

Appendix B Generator Matrices of Optimal Codes

The generator matrices of five instances optimal linear codes are detailed as follows.

- The $(4,2)$-optimal codes with TOB $\mathcal{B}_0$:

$$\mathcal{G}_{D_1} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0 \\ 0&1&0&0&0&0&0&0&1&1&1&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&1&1&0&1&0&1&1&0 \\ 0&0&0&1&0&0&0&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&0&1&1&1&0&1 \\ 0&0&0&0&0&1&0&0&1&0&1&0&1&0&1&1 \\ 0&0&0&0&0&0&1&0&0&1&1&0&0&1&0&1 \\ 0&0&0&0&0&0&0&1&0&0&0&0&1&1&1&1 \end{pmatrix}$$

- The $(4,3)$-optimal codes with TOB $\mathcal{B}_1$:

$$\mathcal{G}_{D_2} = \begin{pmatrix} 1&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0 \\ 0&1&0&0&0&0&0&0&1&0&0&0&1&0&1&0 \\ 0&0&1&0&0&0&0&0&1&0&1&0&0&1&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&1&1&0&1 \\ 0&0&0&0&1&0&0&0&1&1&0&1&0&1&1&1 \\ 0&0&0&0&0&1&0&0&1&0&1&1&1&1&0&1 \\ 0&0&0&0&0&0&1&0&0&1&0&0&1&0&1&1 \\ 0&0&0&0&0&0&0&1&0&0&0&1&1&1&1&0 \end{pmatrix}$$

- The $(4,4)$-optimal codes with the basis $\mathcal{B}_2$:

$$\mathcal{G}_{D_3} = \begin{pmatrix} 1&0&0&0&0&0&0&0&1&1&1&1&1&0&1&1 \\ 0&1&0&0&0&0&0&0&1&1&0&0&0&1&0&1 \\ 0&0&1&0&0&0&0&0&1&1&0&1&1&0&1&0 \\ 0&0&0&1&0&0&0&0&0&1&1&0&1&1&0&1 \\ 0&0&0&0&1&0&0&0&1&0&0&0&1&1&1&0 \\ 0&0&0&0&0&1&0&0&0&1&0&0&0&1&1&1 \\ 0&0&0&0&0&0&1&0&1&0&0&1&1&0&1&1 \\ 0&0&0&0&0&0&0&1&1&1&1&1&0&1&0&1 \end{pmatrix}$$

- The $(4,1)$-optimal binary codes by random draws over $\mathbb{F}_2$:

$$\mathcal{G}_{D_4} = \begin{pmatrix} 1&0&0&0&0&0&0&0&1&0&1&0&0&1&1&1 \\ 0&1&0&0&0&0&0&0&0&1&1&1&0&0&0&1 \\ 0&0&1&0&0&0&0&0&1&0&1&1&0&1&0&0 \\ 0&0&0&1&0&0&0&0&0&1&1&0&1&1&1&1 \\ 0&0&0&0&1&0&0&0&0&0&1&1&1&1&0&0 \\ 0&0&0&0&0&1&0&0&1&1&0&0&1&1&1&0 \\ 0&0&0&0&0&0&1&0&1&1&1&1&0&1&0&0 \\ 0&0&0&0&0&0&0&1&1&1&1&0&1&0&1&1 \end{pmatrix}$$

- The $(5,24)$-optimal binary codes from Magma BKLC database:

$$\mathcal{G}_{D_5} = \begin{pmatrix} 1&0&0&0&0&0&0&0&1&0&0&1&1&1&1&0 \\ 0&1&0&0&0&0&0&0&0&1&0&0&1&1&1&1 \\ 0&0&1&0&0&0&0&0&1&1&0&0&1&1&0&0 \\ 0&0&0&1&0&0&0&0&0&1&1&0&0&1&1&0 \\ 0&0&0&0&1&0&0&0&0&0&1&1&0&0&1&1 \\ 0&0&0&0&0&1&0&0&1&1&1&1&0&0&1&0 \\ 0&0&0&0&0&0&1&0&0&1&1&1&1&0&0&1 \\ 0&0&0&0&0&0&0&1&1&1&0&1&0&1&1&1 \end{pmatrix}.$$

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] Koichi Betsumiya. Minimum Lee weights of Type II codes over $\mathbb{F}_{2^r}$. *Discret. Math.*, 308(14):3018–3022, 2008.

[4] Raphaël Le Bidan, Ramesh Pyndiah, and Patrick Adde. Some Results on the Binary Minimum Distance of Reed-Solomon Codes and Block Turbo Codes. In *Proceedings of IEEE International Conference on Communications, ICC 2007, Glasgow, Scotland, UK, 24-28 June 2007*, pages 990–994. IEEE, 2007.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] Wei Cheng, Sylvain Guilley, and Jean-Luc Danger. Information Leakage in Code-Based Masking: A Systematic Evaluation by Higher-Order Attacks. *IEEE Trans. Inf. Forensics Secur.*, 17:1624–1638, 2022.

[11] Wei Cheng, Yi Liu, Sylvain Guilley, and Olivier Rioul. Attacking Masked Cryptographic Implementations: Information-Theoretic Bounds. In *IEEE International Symposium on Information Theory, ISIT 2022, Espoo, Finland, June 26 - July 1, 2022*, pages 654–659. IEEE, 2022.

[12] Wei Cheng, Yi Liu, Sylvain Guilley, and Olivier Rioul. Towards Finding Best Linear Codes for Side-Channel Protections. In Ulrich Kühne and Fan Zhang, editors, *Proceedings of 10th International Workshop on Security Proofs for Embedded Systems*, volume 87 of *EPiC Series in Computing*, pages 83–99. EasyChair, 2022.

[13] Thomas De Cnudde, Begül Bilgin, Benedikt Gierlichs, Ventzislav Nikov, Svetla Nikova, and Vincent Rijmen. Does Coupling Affect the Security of Masked Implementations? In Sylvain Guilley, editor, *Constructive Side-Channel Analysis and Secure Design - 8th International Workshop, COSADE 2017, Paris, France, April 13-14, 2017, Revised Selected Papers*, volume 10348 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2017.

[14] 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.

[15] Philippe Delsarte. The Association Schemes of Coding Theory. In *Combinatorics*, pages 143–161. Springer, 1975.

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

[17] 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.

[18] Jérôme Lacan and Emmanuelle Delpeyroux. The q-ary Image of Some q^m-ary Cyclic Codes: Permutation Group and Soft-decision Decoding. *IEEE Trans. Inf. Theory*, 48(7):2069–2078, 2002.

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

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] Patrice Rabizzoni. Relation Between the Minimum Weight of a Linear Code over $GF(q^m)$ and Its q-art Image over GF(q). In Gérard D. Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, volume 388 of *Lecture Notes in Computer Science*, pages 209–212. Springer, 1988.

[27] Charles T. Retter. Gaps in the Binary Weight Distributions of Reed-Solomon Codes. *IEEE Trans. Inf. Theory*, 38(6):1688–1697, 1992.

[28] 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.

[29] Katsumi Sakakibara and Masao Kasahara. On the Minimum Distance of a q-ary Image of a q^m-ary Cyclic Code. *IEEE Trans. Inf. Theory*, 42(5):1631–1635, 1996.

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

[31] Richard C. Singleton. Maximum Distance q-nary Codes. *IEEE Trans. Information Theory*, 10(2):116–118, 1964.

[32] University of Sydney (Australia). Magma Computational Algebra System. http://magma.maths.usyd.edu.au/magma/, Accessed on 2022-08-22.

[33] 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.

[34] Isaac Woungang, Alireza Sadeghian, and William W. Melek. Bounds on the Minimum Distances of a Class of q-ary Images of q^m-ary Irreducible Cyclic Codes. In *Proceedings of the 2004 IEEE International Symposium on Information Theory, ISIT 2004, Chicago Downtown Marriott, Chicago, Illinois, USA, June 27 - July 2, 2004*, page 185. IEEE, 2004.

[35] Qianmei Wu, Wei Cheng, Sylvain Guilley, Fan Zhang, and Wei Fu. On Efficient and Secure Code-based Masking: A Pragmatic Evaluation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(3):192–222, 2022.