IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2021, No. 1, pp. 426–450. DOI:10.46586/tches.v2021.i1.426-450

Redundant Code-based Masking Revisited

Nicolas Costes and Martijn Stam

Simula UiB
Merkantilen (3rd floor)
Thormøhlensgate 53D
N-5006 Bergen, Norway.
{nicolas,martijn}@simula.no

Abstract. Masking schemes are a popular countermeasure against side-channel attacks. To mask bytes, the two classical options are Boolean masking and polynomial masking. The latter lends itself to redundant masking, where leakage emanates from more shares than are strictly necessary to reconstruct, raising the obvious question how well such "redundant" leakage can be exploited by a side-channel adversary.
We revisit the recent work by Chabanne et al. (CHES'18) and show that, contrary to their conclusions, said leakage can—in theory—always be exploited. For the Hamming weight scenario in the low-noise regime, we heuristically determine how security degrades in terms of the number of redundant shares for first and second order secure polynomial masking schemes.
Furthermore, we leverage a well-established link between linear secret sharing schemes and coding theory to determine when different masking schemes will end up with essentially equivalent leakage profiles. Surprisingly, we conclude that for typical field sizes and security orders, Boolean masking is a special case of polynomial masking. We also identify quasi-Boolean masking schemes as a special class of redundant polynomial masking and point out that the popular "Frobenius-stable" sets of interpolations points typically lead to such quasi-Boolean masking schemes, with subsequent degraded leakage performance.

Keywords: Coding Theory · Masking · Shamir Secret Sharing · Success Rate

1 Introduction

Context and challenge. Differential power analysis (DPA) in its simplest form can be regarded as running a maximum likelihood estimator individually on each of the 16 key bytes (for AES-128) of either the first or last round key [HRG14]. Against unprotected blockciphers, DPA is almost always successful in recovering the full key, given sufficiently many traces. Quite how many traces depends on various factors, including how noisy the traces are; an established rule of thumb is that the number needed is inversely proportional to the standard deviation of the noise [MOP07, Eq. (6.9)].

The success of DPA has instigated a multitude of countermeasures, of which *masking* is one of the most popular. In traditional masking schemes of order d, each sensitive variable v is shared into $d+1$ shares $\mathbf{c} = c_1, \ldots, c_{d+1}$ using an encoding. The classical example is Boolean masking, where $v = \sum_{i=1}^{d+1} c_i$. In general, the encoding $\mathbf{c}$ would allow recombination of the sensitive variable v, yet a properly masked implementation will only use so-called gadgets to compute on the shares without ever reconstructing a sensitive variable. In a typical masked blockcipher implementation, the main gadgets required are for secure finite-field multiplication and, often as a subroutine, for refreshing the sharing. For masked schemes and sufficiently noisy leakage, the new rule of thumb is that the number of traces required is exponential in the order d [CJRR99, PR13].

Received: 2020-07-15 Accepted: 2020-09-15 Published: 2020-12-03

To protect against glitches, Prouff and Roche [PR11] introduced polynomial masking. Their (d, n) polynomial masking follows Shamir secret sharing: to mask a sensitive variable v, a random degree d polynomial with constant term v is selected and evaluated in n different non-zero interpolation points, leading to n shares. If the number of shares n exceeds the minimum threshold $d + 1$ needed to reconstruct a sensitive variable, we speak of *redundant* masking. By setting $n = 2d + 1$, Prouff and Roche leverage a classical result from multi-party computation [BGW88] to thwart the effect of $\leq d$ glitches on their masked implementation, providing a security proof in the novel glitch model (an extension of the customary probing model [ISW03]). More recently, Seker et al. [SFRES18] showed how the redundancy of polynomial masking can be used to detect up to $n - 2d - 1$ errors to protect against fault attacks.

Polynomial masking with $n = d + 1$ can be considered as alternative to Boolean masking for the non-redundant case. Unlike Boolean masking, $(d, d + 1)$ polynomial masking is *parameterized*, namely by $\mathcal{S}$, the set of interpolation points: a different selection of these points leads to slightly different masking schemes that may leak slightly more or less [WSY$^+$16]. To speed up the squaring gadget, Roche and Prouff [RP12], refined their masking scheme by restricting to a set $\mathcal{S}$ of interpolation points that is stable under the Frobenius automorphism (essentially, squaring an interpolation point is guaranteed to result in another interpolation point). For $(d, n) \in \{(1, 2), (2, 3)\}$ they compare the leakage profile of non-redundant polynomial masking (for unspecified $\mathcal{S}$) with first and second order Boolean masking, respectively, by considering mutual information of noisy Hamming weight leakage for noise deviation $\sigma \in [0, 4.5]$. They conclude that, for the same degree, Boolean masking is considerably more leaky than polynomial masking [RP12, Figure 3].

Unfortunately, redundant masking schemes are excluded from the comparison above. Thus, it is unclear what happens when *redundant* masking schemes are used, where $n > d + 1$. From an information-theoretic perspective, one would expect that *exploiting all available information is always advantageous*, with the only real caveat that computational complexity might increase. Roughly speaking, more information per trace would mean that fewer traces are needed to recover a key, though the processing of these traces might take longer.

Intriguingly, when Chabanne, Maghrebi, and Prouff [CMP18] recently addressed how redundant polynomial masking leaks, they remark that "observing strictly more than $d + 1$ shares will merely provide the attacker with more noise than information" and they argue and experimentally establish that observing $d + 2$ shares leads to better attacks than any subset of $d + 1$ shares if and only if the signal-to-noise ratio is lower than some bound. Thus, their results go counter to the information-theoretic adage.

Our contribution. We revisit the effect of leakage of redundant masking schemes, rephrasing the research questions posed by Chabanne et al. [CMP18] as follows:

1. How does the choice of public interpolation points influence the effectiveness of side-channel attacks against (d, n) polynomial masking schemes?

2. How does the availability of leakage on redundant shares affect the number of traces needed to mount a successful side-channel attack?

To answer the above two questions, we restrict ourselves to the customary simulated setting, where we consider the single byte output of a first round AES S-box as the sensitive variable and we assume an adversary gets access to noisy Hamming weights on each of the shares used to encode said sensitive variable.

The leakage of a polynomial masking scheme can depend on $\mathcal{S}$, that is the choice of the interpolation points—and to a lesser extent on the choice of field representation [WSY$^+$16]. As we want to ensure that our subsequent investigation into the effects of redundancy is not unduly affected by these choices, in Section 2 we recast polynomial masking as

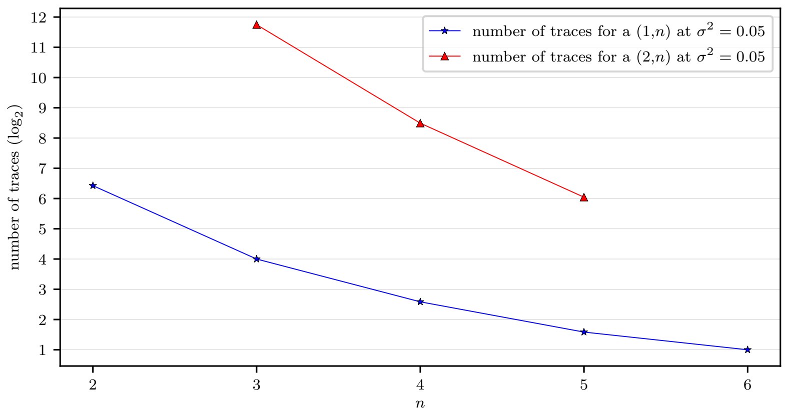

Figure 1: Number of traces needed by ML distinguisher to reach 90% success rate in the noisy Hamming weight model with $\sigma^2 = 0.05$ against redundant (d, n) polynomial masking schemes with $d = 1$ and $d = 2$ for varying n.

linear, code-based masking (Definition 1), which subsequently enables us to formalize (in Section 3.2) the folklore notion of equivalent leakage for masking schemes (Definition 2) and establish when two different interpolation sets are essentially leak-equivalent (Theorem 1).

A more refined investigation (in Section 3.3) reveals that, up to fairly high degree, Boolean masking is equivalent to polynomial masking with appropriately chosen interpolation set. This surprising equivalence highlights even more the importance of interpolation point selection: a particularly poor selection can seriously downgrade security. The same is true for redundant polynomial masking schemes, where we introduce the concept of quasi-Boolean masking (Definition 3). We claim that quasi-Boolean choices have an atypically weak leakage profile and for that reason should, if possible, be avoided. Unfortunately, it turns out that Roche and Prouff's suggestion of Frobenius-stable interpolation sets—also taken up by Seker et al. [SFRES18]—typically leads to quasi-Boolean behaviour.

So far, our claims are primarily *qualitative*: for instance that quasi-Boolean behaviour is detrimental to security. We back this claim up by determining the number of traces needed by the optimal maximum-likelihood (ML) distinguisher to achieve a success rate of 90%. However, before we do so, we expand on the mathematics behind the ML distinguisher in case of redundant masking schemes in Section 4. Although we dare not claim novelty here—all we are doing here is applying the well-known concept of a ML distinguisher to the current context—it is here that Chabanne et al. [CMP18] made a mistake, resulting in their, in retrospect, erroneous conclusions.

With the distinguisher sorted, we first turn our attention to quasi-Boolean and Frobenius masking (Section 5.1). As expected, we confirm that polynomial masking is equivalent to Boolean masking and performs the same as Boolean masking of the same order; moreover, we can see that the quasi-Boolean $(1, 3)$ Frobenius masking performs considerably worse than typical $(1, 3)$ polynomial masking.

We then turn our attention to a wider spectrum of choices of (d, n) for "typical" polynomial masking (Section 5.2), where we use the same set of interpolation points as Chabanne et al. Somewhat surprisingly, we observe that the very low noise $\sigma^2 = 0.05$ case is reasonably representative even for higher noise when comparing different choices of (d, n). Specifically, it is **not** the case that redundant masking performs worse than non-redundant masking or that it is advantageous, as Chabanne et al. [CMP18] claim, to only consider a subset of the shares being leaked upon.

So let's treat the number of traces needed to attain 90% success rate at noise level

$\sigma^2 = 0.05$ as a simple, approximate metric for the hardness of mounting a side-channel attack against typical polynomial masking with parameters (d, n). We plotted this metric for select choices of (d, n) in Figure 1, yielding a useful quantitative insight in *how much faster the MLE key recovery runs (in terms of the traces needed) when the sharing becomes more redundant.*

For $d = 1$ having one redundant share, so moving from $n = 2$ to $n = 3$, implies more than a five-fold reduction in the number of traces needed, whereas two redundant shares (from $n = 2$ to 4) yield a fifteen-fold reduction. After that, the returns of additional redundant shares are diminishing. For $d = 2$ the initial reductions are even more stark: almost tenfold for one redundant share and over fiftyfold for two.

To put these numbers in perspective, imagine a hypothetical adversary that given a single leakage on n shares, creates $\binom{n}{d+1}$ leakages on $d + 1$ shares and runs a maximum likelihood distinguisher, ignoring the dependency introduced in the process. Then the number of traces for $d = 1$ would only increase threefold for $n = 2$ to 3 and sixfold for $n = 2$ to 4. Similarly, for $d = 2$ the gains would 'only' be fourfold and thirtyfold. As we demonstrate, a direct multivariate distinguisher exploiting the redundancy between the shares gains even more!

Related work. Moradi and Mischke [MM13] attack the original polynomial masking scheme [PR11], so with random public evaluation points. They concentrate on the case $d = 1$ and, for their experiments, select points $\{02, 03, 04\}$; this set is not Frobenius stable. They use an experimental hardware setup to mount a successful correlation-collision attack using second moments and about a million traces. Their results are incomparable to ours.

Goubin and Martinelli [GM11] introduced a slightly different version of polynomial masking, where the interpolation points were treated as part of the encoding: initially selected at random, they change during computations (e.g. for squaring or mask refreshing). Thus, the interpolation points are rightly treated as part of the encoding, rather than as a fixed parameter and consequently reconstruction is no longer linear in the encoding (cf. inner product masking, Remark 3). The scheme was subsequently shown to be flawed [CPR12].

To gain confidence that the operations on masked intermediates do not leak, security of masking schemes is often formally analysed in an appropriate probing model [ISW03, BBD+16]. Security in the probing model is information-theoretic and at first sight somewhat removed from practice: after all, an adversary observing traces learns a little about all intermediate values, as opposed to the probing's model concept of learning everything about a few intermediates. Yet under appropriate assumptions, security in the probing model does have implications to the real-world by providing upper bounds on the success rate of an adversary [DDF19, DFS19]. However, the proven bounds are not very sharp and can be vacuous even for relatively high noise regimes.

A typical implementation of d-th order multiplication might easily leak d times on the same share, so counterintuitively, in some models increasing the number of shares might actually decrease security [BCPZ16]. As said, we concentrate on leakage on the individual shares only and ignore how computation might influence the effective SNR or lead to leakage on multiple shares simultaneously. The latter problem has recently also received attention in the context of bit-sliced implementations [GMPO19].

Although masking schemes are well known to relate to secret sharing schemes and secret sharing schemes to coding theory, the coding-theoretic perspective of masking schemes is relatively underexplored. Several prior works linking masking schemes to coding theory establish a direct link between the two concepts [CRZ13, BCC+14, WMCS20], ignoring the link through secret sharing schemes. Yet, as we will demonstrate, exploiting this intermediate link is highly beneficial in understanding the leakage potential of masking schemes. Furthermore, most work concentrates on how to perform computations on masking schemes, code-based or not, and classifies masking schemes primarily in those terms [GSF14]: multiplication gadgets for Boolean are faster than those for polynomial masking with Frobenius-stable interpolation sets (which in turn should be faster than those for arbitrary polynomial masking).

2 Preliminaries

2.1 Masking Schemes and their Leakage

Masking. Masking schemes are commonly used to complicate power analysis attacks. These schemes consist of an encoding mask, where a key or sensitive variable v is represented using multiple, randomized shares by $\mathbf{c} \leftarrow_{\$} \mathsf{mask}(v)$. We write $\mathcal{C}(v)$ for the support of mask on input v, that is $\mathbf{c} \in \mathcal{C}(v)$ iff there is a non-zero probability that $\mathsf{mask}(v)$ returns $\mathbf{c}$. A correct masking scheme satisfies that $\mathcal{C}(v)$ and $\mathcal{C}(v')$ are disjoint for distinct v and v'.

Additionally, a masking scheme requires "gadgets" to perform operations in the encoded domain. These gadgets avoid ever having to reconstruct the key (or any intermediate value that depends deterministically on the key and known inputs), thereby ostensibly reducing the leakage at any point in time and forcing an attacker to attempt more expensive and often less effective higher order attacks instead.

Many different masking schemes have been proposed over the years, varying in both their encodings and how to compute on them. We are exclusively interested in the encoding, i.e. in mask, and then primarily in those that are based on linear codes suitable for secret sharing. This includes Boolean masking [CJRR99], as well as polynomial masking [GM11, PR11] and the closely related revisited inner product masking [BFG15].

Side-Channel Analysis. To analyse the security of a masking scheme, we imagine an adversary trying to recover the unknown key $k \in \mathbb{F}$ based upon multiple, independent leakages $\mathsf{Leak}(k)$ on said key. We will concentrate on the scenario where each leakage is on a relevant S-box output, so that technically the leakage consists of a randomly chosen plaintext x_i and a leakage trace on a sensitive variable, in our case an S-box output on $k \oplus x_i$.

Furthermore, we adopt the common scenario where for each trace, the sensitive variable is freshly masked and the resulting shares each individually and independently "drip" some leakage, i.e. for each share c_j we obtain a noisy observation of some deterministic transformation $f(c_j)$. Figure 2 formalizes the full key-recovery under leakage setting, where we further narrowed down to dripping the Hamming weight with independent Gaussian noise of variance σ^2. Thus we are in the well-trodden noisy Hamming weight model (and see below).

We use the notation $\mathcal{N}(0, \sigma^2)$ to denote the drawing of this noise; later we will use $\mathcal{N}\left(x \mid \mu, \sigma^2\right)$ for evaluating the probability density function (pdf) of a Gaussian with mean μ and variance σ^2 in x. Here we prefer the use of σ^2 over σ or $\log_{10} \sigma$ to have a more direct link with the signal-to-noise ratio (SNR), but obviously one can easily move to and fro. Related to SNR, for Hamming weight leakage on a uniform variable in $\mathbb{F}_{2^8}$, the signal variance is 2, so the SNR is $2/\sigma^2$.

Typically there are three interesting regimes of noise to consider: the low noise scenario where behaviour is governed primarily by the noiseless scenario, the high noise scenario where behaviour is starting to follow the asymptotic trend, and the medium noise scenario to bridge the change in behaviour. We interpret behaviour here as the key-recovery success rate of an adversary. Bruneau et al. [BGHR14] considered $\sigma^2 = 1$ still as low noise, whereas $\sigma^2 = 9$ was called high; Cheng et al. [CGC⁺20] recently indicated the bounds $\sigma^2 \leq 2^{-1}$ for the low noise case and $\sigma^2 \geq 2$ for the high noise case.

```
$\mathsf{Exp}^{\mathrm{kr(n)}}_{(n,d)}(\mathbb{A})$
$k \leftarrow_{\$} \mathbb{F}$
for $i \in \{1, \dots, N\}$
    $\ell_i \leftarrow \mathsf{Leak}(k)$
$\hat{k} \leftarrow \mathbb{A}(\ell_1, \dots, \ell_N)$
return $\hat{k} = k$
```

```
$\mathsf{Leak}(k)$
$x_i \leftarrow_{\$} \mathbb{F}$
$v_i \leftarrow \mathtt{S}(k \oplus x_i)$
$\mathbf{t}_i \leftarrow \mathsf{Trace}(v_i)$
return $(x_i, \mathbf{t}_i)$
```

```
$\mathsf{Trace}(v_i)$
$\mathbf{c} \leftarrow \mathsf{mask}(v_i)$
for $j \in \{1, \dots, n\}$
    $t_{ij} \leftarrow_{\$} \mathsf{Drip}(c_j)$
return $(t_{i1}, \dots, t_{in})$
```

```
$\mathsf{Drip}(c_j)$
return
    $\mathrm{Hw}(c_j) + \mathcal{N}(0, \sigma^2)$
```

Figure 2: The key recovery game given noisy Hamming weight leakage of a masked `S`-box output.

The noisy Hamming weight model has been used extensively, including for the study of masking schemes [RP12, GM11, BFG15, PGS$^+$17] and the simulation of leakage [TAL09, dHVdV$^+$03]. Yet, it is good to realize its limitations as a model. In a practical setting, traces are acquired via measuring the power consumption of the chip or the eletromagnetic radiations it emits: each measurement yields a leakage trace where several points of interest (or features) can be considered (resp. extracted), depending on which operation is targeted.

The Hamming weight model and the Hamming distance model are historically the most popular "simple" models and Mangard, Oswald, and Popp [MOP07, Section 3.3.2] argue that, in most cases, the Hamming distance is the more appropriate of the two as it better captures transitions. Recent investigations into the Cortex-M0 [MOW17] confirmed that most operations leak on transitions, moreover not all bits (of a byte) contribute equally to the leakage produced. Only for the `load` operation could the Hamming weight model be considered appropriate. In a similar vein, Kannwischer et al. [KPP20] showed that for an 8-bit AVRXMEGA (as well as for a 32-bit STM32F405) the `store` operation also leaks Hamming weight, with $\sigma \approx 0.5$ (for the AVRXMEGA). They conclude that, when considering interaction with the SRAM, Hamming weight leakage is a good model. Poussier et al. [PGS$^+$17] investigate an implementation that performs successive `load` or `store` on all the shares of the output of the first SBOX and also observe Hamming weight leakage.

Notwithstanding these results supporting the Hamming weight model, for most implementations said model is probably not a good representation of reality. For instance, a bitsliced or n-sliced implementation would not exhibit such leakage. All in all, when a specific masked implementation on a particular platform is considered (for security evaluation), suitably refined models are likely more appropriate, ranging from the Hamming distance model, from the weighted Hamming weight model, to a more advanced leakage emulator such as ELMO [MOW17].

Our main motivation to opt for the noisy Hamming weight model are its simplicity when running simulations on a range of σ^2 and its popularity in the literature, facilitating comparison with previous work. Our results, expressed in the number of traces needed to achieve a certain success rate, should therefore not be thought of as a proxy for real attacks in any scenario, but rather as a means to compare the effect of different masking design choices on the potential to leak, in several noise regimes.

Mutual Information. When regarding the key-recovery experiment (Figure 2), in first instance we are interested in the success rate as a function of the number of traces N, where success rate is simply the probability that indeed $\hat{k} = k$ at the end. Conversely, we can also look at the number of traces N needed to achieve a given success rate. We will focus on the latter metric, typically for a success rate of 90%.

A very popular alternative to looking at success rates or trace-complexity directly is to evaluate the mutual information $I(K; L_1)$ between the random key variable K (corresponding to the choice of k in our experiment) and the leakage L_1 on it given a single "leak", thus corresponding to ℓ_1 in our experiment. The number of traces needed to achieve a given success rate is believed to be correlated to the reciprocal of this mutual information [SVO+10a]. We believe that looking directly at success rates through simulated attacks, though computationally more costly, provides a more precise picture when comparing different masking choices. Indeed, as we will see, in some cases we *know* that two masking schemes will lead to the same mutual information, yet we cannot prove that they lead to identical success rate curves.

2.2 Linear Codes

An $[\bar{n}, \bar{k}, \bar{d}]_{\mathbb{F}}$ linear error-correcting code $\mathcal{C}$ is the set of elements (codewords) in a $\bar{k}$-dimensional subspace of $\mathbb{F}^{\bar{n}}$, where $\mathbb{F}$ is a finite field and the minimum distance $\bar{d}$ is defined as the minimum Hamming weight, taken over all nonzero codewords in $\mathcal{C}$. Below we list some elementary facts about linear error-correcting codes relevant to this work; more details can be found in any of the standard textbooks [vL99, MS77]. As an aside, we use bars for $\bar{n}, \bar{k}, \bar{d}$, etc. to detangle a notational knot later on.

An $[\bar{n}, \bar{k}, \bar{d}]_{\mathbb{F}}$ code $\mathcal{C}$ can be generated by a matrix $G \in \mathbb{F}^{\bar{k}\times\bar{n}}$, meaning that $\mathcal{C} = \{\bar{\mathbf{x}} \cdot G | \bar{\mathbf{x}} \in \mathbb{F}^{\bar{k}}\}$, using row vectors throughout. If G is a generator of an $[\bar{n}, \bar{k}, \bar{d}]_{\mathbb{F}}$ code and $A \in \mathbb{F}^{\bar{k}\times\bar{k}}$ is invertible, then $G' = A \cdot G$ generates the same code. For an index set $\mathcal{I} \subseteq \{1, \dots, \bar{n}\}$ we define $G_{\mathcal{I}} \in \mathbb{F}^{\bar{k}\times|\mathcal{I}|}$ as the restriction of G to those columns indexed by $\mathcal{I}$ and similarly $\bar{\mathbf{c}}_{\mathcal{I}}$ as the restriction of $\bar{\mathbf{c}} \in \mathcal{C}$ to those positions indexed by $\mathcal{I}$. If $A = G_{\mathcal{I}}$ for $\mathcal{I} = \{1, \dots, \bar{k}\}$ is invertible, then $G' = A \cdot G$ is *systematic*, meaning $G' = [I_{\bar{k}}|P]$ with $I_{\bar{k}} \in \mathbb{F}^{\bar{k}\times\bar{k}}$ the identity matrix and $P \in \mathbb{F}^{\bar{k}\times(\bar{n}-\bar{k})}$.

Two codes are *equivalent* iff one can obtain one code from the other by permuting the positions of all codewords. For reasons explained later, we deliberately do *not* include position-wise scalar multiplication in our definition of equivalence. We call two generator matrices G and G' resulting in equivalent codes *equivalent.* Two generator matrices are equivalent iff there exist invertible $A \in \mathbb{F}^{\bar{k}\times\bar{k}}$ and permutation matrix $B \in \mathbb{F}^{\bar{n}\times\bar{n}}$ such that $G' = A \cdot G \cdot B$.

An important class of codes are so-called maximum distance separable (MDS) codes. These codes satisfy $\bar{n} - \bar{k} = \bar{d} - 1$, which is optimal in the sense the Singleton bound $\bar{n} - \bar{k} \geq \bar{d} - 1$ is met. For our purposes, the minimum distance $\bar{d}$ is not that relevant and we will henceforth drop it from our notation. We are interested in the special properties of an MDS code's generator matrix, specifically, G is the generator matrix of an MDS code if and only if any $\bar{k}$ columns are linearly independent. Thus, if $\mathcal{I} \subseteq \{1, \dots, \bar{n}\}$ with $|\mathcal{I}| = k$, then $G_{\mathcal{I}}$ is invertible. Consequently, the dual code $\mathcal{C}^{\perp}$ of an $[\bar{n}, \bar{k}]_{\mathbb{F}}$ MDS code $\mathcal{C}$ is an $[\bar{n}, \bar{n} - \bar{k}]_{\mathbb{F}}$ MDS code, where by definition the dual code $\mathcal{C}^{\perp}$ is the vector space of $\mathbb{F}^{\bar{n}}$ orthogonal to $\mathcal{C}$, i.e. $\bar{\mathbf{d}} \in \mathcal{C}^{\perp}$ iff $\bar{\mathbf{c}} \cdot \bar{\mathbf{d}}^T = 0$ for all $\bar{\mathbf{c}} \in \mathcal{C}$.

The most famous class of MDS codes are Generalized Reed–Solomon (GRS) codes, based on polynomial evaluation: each polynomial of degree $< \bar{k}$ defines a codeword of length $\bar{n}$ by evaluating the polynomial in $\bar{n}$ distinct elements α_i of the field $\mathbb{F} = \mathbb{F}_q$, followed by multiplication by a coordinate-wise constant β_i. The resulting generator matrix is

$$G = \begin{pmatrix} \beta_1 & \beta_2 & \dots & \beta_n \\ \beta_1\alpha_1 & \beta_2\alpha_2 & \dots & \beta_n\alpha_n \\ \vdots & & & \\ \beta_1\alpha_1^{\bar{k}-1} & \beta_2\alpha_2^{\bar{k}-1} & \dots & \beta_n\alpha_n^{\bar{k}-1} \end{pmatrix}$$

where the column $(0, \dots, 0, \beta)^T$ is also allowed, corresponding to evaluation in the point at infinity, so $\alpha = \infty$ (cf. [CDN15, Section 11.7]). Thus, for $q = |\mathbb{F}|$ and all $1 \leq \bar{k} \leq q + 1$ there exists a $[q + 1, \bar{k}]_{\mathbb{F}}$ GRS hence MDS code [MS77, Ch. 4, Theorem 9].

3 The Theory of Code-based Masking

3.1 Secret Sharing and Masking

In 1979, Blakley [Bla79] and Shamir [Sha79] concurrently introduced the concept of secret sharing (see [CDN15, Chapter 11] for a modern treatise). We are exclusively interested in threshold schemes. Here a dealer shares a secret amongst n participant in such a way that only subsets of strictly more than d participants are capable of reconstructing the subset, yet unauthorized subsets (of at most d participants) cannot learn anything about the secret whatsoever. Massey [Mas93] showed a general transform from any $[n+1, d+1]$ MDS code to a secret sharing scheme with n players and privacy threshold d: given the code $\mathcal{C}$ and a fixed position $i \in \{1, \ldots, n+1\}$ in the code, to share a secret $s \in \mathbb{F}$ randomly pick a codeword $\bar{\mathbf{c}} \in \mathcal{C}$ satisfying $c_i = s$. The remaining positions (differing from i) of the codeword make up the n shares.

Threshold schemes have been suggested as a countermeasure against side-channel attacks (esp. DPA) under the name masking, initially for the special case of d-order masking where $n = d+1$, but later also for the redundant case $n > d+1$. Here the order d (for the masking scheme) corresponds to the privacy threshold of the associated secret sharing scheme; probing security up to d-probes immediately follows from this privacy threshold. Masking schemes based on linear secret sharing schemes can most easily be expressed by exploiting the link with coding theory.

Definition 1 (Code-based Masking)**.** Let G be a generator for a $[n+1, d+1]_{\mathbb{F}}$ MDS code with first column $(1\ 0\ \ldots 0)^T$, then to mask a sensitive variable $v \in \mathbb{F}$, select $u_i \leftarrow_{\$} \mathbb{F}$ uniformly at random for $i \in \{1, \ldots, d\}$ and calculate $\bar{\mathbf{c}} \leftarrow (v\ u_1\ \ldots u_d) \cdot G$. Indexing $\bar{\mathbf{c}}$ from zero, so $\bar{\mathbf{c}} = (c_0, c_1, \ldots, c_n)$, share i equals c_i, for $i \in \{1, \ldots, n\}$. We call $\mathbf{c} = (c_1, \ldots, c_n)$ a code-based masking of v and may write $\mathbf{c} \leftarrow_{\$} \mathsf{mask}_G(v)$.

Remark 1. Sticking to MDS codes ensures both that any $d+1$ shares suffice to reconstruct the sensitive variable v yet that no d shares jointly provide any information about v. The first column of G being $(10\ldots0)$ ensures $c_0 = v$, corresponding to Massey's construction using the first position. If G is systematic, then $c_i = u_i$ for $i \in \{1, \ldots, d\}$ which matches the traditional use of the word "mask" in the literature. If furthermore $n = d+1$ then the final c_{d+1} is known as the masked variable; for redundant masking, where $n > d+1$, this terminology may become misleading.

Lemma 1 (Code-based Reconstruction)**.** *Let G be a generator for a $[n+1, d+1]_{\mathbb{F}}$ MDS code $\mathcal{C}$ with first column $(10\ldots0)^T$, and let $\mathbf{c} \leftarrow_{\$} \mathsf{mask}_G(v)$ be a code-based masking of some v. Let $\bar{\mathbf{d}} \in \mathcal{C}^{\perp}$ indexed from zero, so $\bar{\mathbf{d}} = (d_0, \ldots, d_n)$, that satisfies $d_0 = -1$. Let $\mathbf{d} = (d_1, \ldots, d_n)^T$, called a reconstruction vector. Then $v = \mathbf{c} \cdot \mathbf{d}$.*

Proof. By construction of code-based masking, we can uniquely extend $\mathbf{c}$ to $\bar{\mathbf{c}} = (v|\mathbf{c}) \in \mathcal{C}$. Then by definition of the dual code $\mathcal{C}^{\perp}$, $0 = \bar{\mathbf{c}} \cdot \bar{\mathbf{d}}^T = -v + \mathbf{c} \cdot \mathbf{d}^T$. □

Remark 2. Recall that the dual of an $[n+1, d+1]_{\mathbb{F}}$ MDS code itself is an $[n+1, n-d]$ MDS code. Thus, for non-redundant schemes with $n = d+1$ it follows that $n - d = 1$, implying a *unique* reconstruction vector $\mathbf{d}$, with no zero coordinates. However, for redundant schemes, $n - d > 1$ and reconstruction vectors are no longer unique.

The simplest meaningful example, for $n = d+1$, is an analogue of the one-time pad. Given a sensitive variable $v \in \mathbb{F}$, select random shares $c_1, \ldots, c_n \in \mathbb{F}$ subject to

$$v = \sum_{i=1}^{n} c_i = c_1 \oplus \ldots \oplus c_n \ ,$$

where the $\oplus$-notation is indicative of the typical cryptographic case of a binary finite field, so field addition corresponds to bitwise exclusive or. The corresponding masking scheme is commonly referred to as d-th order Boolean masking (where $d = n - 1$). It corresponds to code-based masking with generator matrix $G_{\text{bool}} = [I_{d+1}|\mathbf{1}_{d+1}]$ where $\mathbf{1}_{d+1}$ is the all-1 column vector in $\mathbb{F}^{d+1}$.

For the more general threshold case, $n \geq d+1$, Shamir secret sharing is the best known. Given a sensitive variable $v \in \mathbb{F}$, select a random polynomial $p(x)$ over $\mathbb{F}$ of degree at most d such that $p(0) = v$. Then the shares are evaluations of $p(x)$ in points differing from 0, where customarily player i gets share $p(\alpha_i)$. We refer to the set $\mathcal{S} = \{\alpha_1, \ldots, \alpha_n\}$ as the interpolation set. Any $d + 1$ players can uniquely reconstruct the polynomial and retrieve the secret, e.g. using Lagrange interpolation. On the other hand, for any d players, each secret is still equally likely. Shamir secret sharing can be derived from generalized Reed–Solomon codes by setting $\beta_i = 1$ for all i [MS81] and $\alpha_0 = 0$ (after re-indexing the columns). For the resulting masking scheme, which we will refer to as polynomial masking, it is still possible that $n = d + 1$, but we will also study the redundant $n > d + 1$ case.

Revisited inner product (RIP) masking works exclusively for the case $n = d + 1$. First fix a public reconstruction vector $\mathbf{d} \in (\mathbb{F}^*)^n$ subject to $d_n = 1$. To share v, select a random $\mathbf{c}$ satisfying $v = \mathbf{c} \cdot \mathbf{d}^T$, typically by first selecting $c_1, \ldots, c_d$ uniformly at random and then solving for c_n. In that case, the generator matrix is

$$G_{\text{rip}} = \begin{pmatrix} 1 & \mathbf{0} & 1 \\ \mathbf{0} & I_d & -\mathbf{d}^T \end{pmatrix} .$$

Remark 3. Inner product masking as originally devised [BFGV12, DF12] masked a sensitive variable v by selecting *both* $\mathbf{d}$ and $\mathbf{c}$ at random subject to correct reconstruction $v = \mathbf{c} \cdot \mathbf{d}^T$ (where $\mathbf{d} \in (\mathbb{F}^*)^n$ but without the $d_n = 1$ constraint), considering $(\mathbf{c}, \mathbf{d})$ as the masking. Proper inner product masking is easily seen *not* to be linear as the reconstruction formula is clearly of degree 2. Balasch et al. [BFGV12] already observed that a non-redundant polynomial masking scheme would emerge by fixing $\mathbf{d}$ to the appropriate Lagrange coefficients. Later, Balasch et al. [BFG15] suggested to fix $\mathbf{d}$ as described above, with $d_1 = 1$ (we found fixing $d_n = 1$ easier on notation), giving rise to RIP (which is of course linear). Confusingly, RIP is often referred to as inner product masking (IPM), ignoring the rather crucial difference between linear and non-linear schemes (cf. the diffence between polynomial masking with fixed versus flexible interpolation points [GM11, PR11]).

Remark 4. For comparison, Wang et al. [WMCS20] use a slightly different formalization of code-based masking. They allow packed secret sharing and they omit the zero-indexed column that we use to ensure the sensitive variable can be recovered (instead they impose appropriate rank conditions on the various matrices involved). The difference is clearly visible by comparing their Figure 2 with our G_{bool} and G_{rip}. Packed secret sharing does have the advantage that it is easier to consider a masking scheme at the bit-level, by explicitly mapping $\mathbb{F}_{2^e}$ to $\mathbb{F}_2^e$. We ignore the choice of the finite field representation, which is not without loss of generalization as Hamming weight leakage does actually depend on this choice. We settle for the standard AES field representation throughout.

3.2 Equivalence of Code-based Masking Schemes

What is equivalence? Even when fixing the security order d, the number of shares n, and the finite field $\mathbb{F}$, code-based masking is parameterized by a generator matrix G. Some generators lead to identical or equivalent codes, but does this also imply that the corresponding code-based masking schemes "leak equivalent"? To answer this question, we first need to pin down what equivalence of leakage entails.

Consider again the key recovery game (Figure 2) and notice the adversarial input $\ell_1, \ldots, \ell_N$. Over the randomness of the game, we can consider these inputs as random variables, which we will denote $L_1, \ldots, L_N$ respectively. The distribution of these random variables depends on a number of choices, namely the S-box, the Drip-function, and finally the masking scheme (plus implicitly the finite field representation). We already fixed the Drip-function to noisy Hamming weight (parameterized by σ^2), the S-box to the AES one, and the finite field representation to the customary AES one. This leaves the masking scheme or, for code-based masking, the choice of the generator matrix. Suppose G and G' are both suitable for CBM, then denote with $L^{(N)} = (L_1, \ldots, L_N)$ the random variable for the leakage induced by mask_G and with $L'^{(N)} = (L'_1, \ldots, L'_N)$ that by $\mathsf{mask}_{G'}$. The leakages are equivalent if anything an adversary can do with $L^{(N)}$ it can also do with $L'^{(N)}$ and vice versa.

Definition 2 (Leak-equivalence)**.** Let n and d be given. Two generators G and G', both in $\mathbb{F}$, are *fully leak-equivalent* iff there exists a bi-efficient bijection π such that the distributions $L^{(N)}$ and $\pi(L'^{(N)})$ are identical for all N. They are *leak-equivalent* iff there exists a bi-efficient bijection π such that the distributions L_1 and $\pi(L'_1)$ are identical.

Remark 5. As we tailored our notation in Figure 2 to our later experimental setting, we defined leak-equivalence in those terms as well. Of course, it is easy to generalize the concept by allowing different kind of Leak functions. We require the bijection to be efficiently computable in both directions. We refrain from providing a formal computational model, but rely on an intuitive understanding of efficiency; in our context efficiency should always be rather obvious and incontrovertible.

General relationships. Fully leak-equivalence immediately implies leak-equivalence. A natural question is whether the converse is true, but as we will see shortly, there are strong arguments why this is unlikely. Leak-equivalence is as least as strong as having identical mutual information, as we formalize in the lemma below.

Lemma 2 (Relationship with mutual information)**.** *Leak-equivalence implies $I(K; L_1) = I(K; L'_1)$ and full leak-equivalence implies $I(K; L^{(N)}) = I(K; L'^{(N)})$ for all N.*

Proof. The implications are straightforward and can be seen as a consequence of both $K \to L^{(N)} \to L'^{(N)}$ and $K \to L'^{(N)} \to L^{(N)}$ forming Markov chains so the data-processing inequality implies both $I(K; L^{(N)}) \geq I(K; L'^{(N)})$ and $I(K; L'^{(N)}) \geq I(K; L^{(N)})$. □

Suppose that we would consider a slightly different Leak-function, where the sensitive variable v_i equals the key k to be recovered (essentially always pick $x_i \leftarrow 0$ and use the identity S-box). For this Leak-function, leak-equivalence implies fully leak-equivalence. Moreover, if we restrict to Boolean masking and noiseless Hamming-weight leakage (on the shares), we know that even with $N \to \infty$, we only ever learn the Hamming weight of the key. Thus the success rate will not tend to 1. Let's call this the deterministic scenario.

Now keep the masking and per-share leakage the same, but reintroduce the random selection of x_i and a decent S-box, i.e. we are back in our normal scenario. Then it's easy to see—as has been seen before—that the single trace mutual information $I(K; L_1)$ is the same in the normal and the deterministic scenario. In the normal scenario given enough traces the key will roll out (e.g. [LPR+14]), so $\mathrm{H}\left(K \mid L^{(N)}\right)$ tends to zero. Yet in the deterministic scenario, uncertainty remains, so $\mathrm{H}\left(K \mid L^{(N)}\right)$ cannot tend to zero. Consequently, having identical $I(K; L_1)$ cannot imply full leak-equivalence.

The code-based case. Let's consider when two code-based masking schemes are (fully) leak-equivalent. We start with the easier case, where two generator matrices produce equivalent but not necessarily identical codes, yet the matrices are closely related.

Lemma 3. *Let G be a (d, n) CBM-suitable generator and let B be a permutation matrix on $n + 1$ elements, corresponding to permutation π on the set $\{0, \ldots, n\}$. Assume $\pi(0) = 0$, then $G' = G \cdot B$ is (d, n) CBM-suitable and fully leak-equivalent to G.*

Proof. The permutation π simply shuffles the shares around and since leakage is i.i.d. for the shares (through Drip), it is easy to shuffle the leakage accordingly. $\square$

The case where two distinct generators produce identical codes turns out a bit more tricky and we can only prove leak-equivalence under the extra assumption of uniformly chosen keys.

Theorem 1. *Let G and G' be (d, n) CBM-suitable generators defining an identical code $\mathcal{C}$. Then for uniform secrets, G and G' are leak-equivalent.*

Proof. A code-based masking scheme defines parallel affine spaces, as well as a bijection from each sensitive variable v to the corresponding affine space $\mathcal{C}(v)$. (Linearity ensures that $v = 0$ is mapped to an actual subspace and if $n - d = 2$ then the affine spaces are in fact hyperplanes that partition the full space.) To mask a sensitive variable, a random element from $\mathcal{C}(v)$ is selected.

If G and G' generate identical codes, the affine spaces will be the same but the mappings from sensitive variables to affine space might differ. As we assume the key is chosen uniformly at random (and the S-box in Leak is a permutation), even conditioned on X_1, the sensitive variable being masked is uniformly random as well. Thus over the choice of both the key and the masking, a codeword is selected uniformly from the full code $\mathcal{C}$, irrespective of the generator being used and even when conditioned on X_1.

As the trace is calculated directly on the codeword (without further reference to other variables), identical codeword distributions imply an identical trace distribution. $\square$

As a corollary, if Leak is deterministic, then uniform keys and generators for the same code lead to full leak-equivalence. Effectively, each $\mathcal{C}(v)$ induces its own leakage distribution over $\mathbb{R}^n$ (cf. the heat maps for second order DPA [SVO+10b, Figs. 11-13]). Irrespective of the generator matrix, the uniform choice of the key leads to the uniform selection of one the affine spaces $\mathcal{C}(v)$, and subsequently leakage follows the corresponding distribution over $\mathbb{R}^n$.

Remark 6. In the above scenario, the uniformity of the keys can also seen to be necessary. Some distributions are closer to each other than others, e.g. in terms of Jensen–Shannon divergence. As an example of a non-uniform distribution, suppose only two keys are possible from the much larger $\mathbb{F}$. The choice of generator matrix, even for identical codes, determines how our two keys are mapped to different $\mathcal{C}(v)$s and, hence, to different distributions over $\mathbb{R}^n$. A generator selecting two very close distributions for the two keys will obviously be less leaky than a generator that ends up assigning two very remote distributions for those same keys.

Remark 7. If we take into account plaintext whitening and the S-box, then each x_i together with the S-box and generator matrix, fixes a a new bijection between keys and affine spaces. If we could select those x_i anyway we want, we get an impression why proving full leak-equivalence is tricky. Imagine we are working over $\mathbb{F}_2^2$ so there are only four affine planes to consider. Suppose that, for some magical reason, the induced leakage distribution for $\mathcal{C}(v_0 v_1)$ is a 2-dimensional Gaussian centered around $(v_0, v_1) \in \mathbb{R}^2$. Taking one step further into the rabbit hole, our choice of S and x_is is such that generator G always assigns key k to sensitive variable $v = k$, whereas generator G' the mapping will depend on i. Then the key recovery advantages under G and G' differ. Technically, the above argument leaves open the possibility that full leak-equivalence can be proven. For instance, in the real game the x_i are chosen at random, yet they are known to the adversary which hampers exploiting said randomness in a proof setting. For most cases, we expect that leak-equivalence leads to identical or at least very close success rates for arbitrary N.

3.3 Boolean and Quasi-Boolean Masking

Boolean masking is often treated as distinct from polynomial masking. However, when we consider polynomial masking as being parameterized by an interpolation set $\mathcal{S}$, some of these sets might lead to polynomial masking leak-equivalent to Boolean masking. An immediate, necessary condition for an equivalent polynomial masking scheme to exist is that there exist $d+1$ different interpolation points in the field $\mathbb{F}_{2^e}$, including the point at infinity and excluding 0. Thus $d+1 \leq 2^e$ is a hard requirement. In Lemma 4 we set out two further necessary conditions for equivalence, phrased in terms of properties of the interpolation set.

Lemma 4. *Consider d-order Boolean masking over $\mathbb{F}_{2^e}$ and let $\mathcal{S}$ be an interpolation set for an equivalent polynomial masking scheme. Then $\infty \in \mathcal{S}$ iff d odd and $\sum_{s\in\mathcal{S}\setminus\{\infty\}} s = 0$ if $d > 1$ and $\mathcal{S} = \{1, \infty\}$ if $d = 1$.*

Proof. Let G be a generator matrix for a CBM scheme. Then it is equivalent to Boolean masking iff $\mathbf{1}$ is in the dual code, so $G \cdot \mathbf{1} = \mathbf{0}$. Thus each row of G has to sum to 0.

The first row sums to $1 + \sum_{s\in\mathcal{S}\setminus\{\infty\}} 1$, which implies that the number of elements in $\mathcal{S}$ excluding ∞ has to be odd. As the total number of elements in $\mathcal{S}$ equals $d+1$, ∞ has to be included in $\mathcal{S}$ iff $d+1$ is even, or equivalently iff d is odd.

The second row sums to $b + \sum_{s\in\mathcal{S}\setminus\{\infty\}} s$ where $b = 1$ iff $\infty \in \mathcal{S} \wedge d = 1$ and $b = 0$ otherwise. If $d = 1$, then $\infty \in \mathcal{S}$ and as $\mathcal{S}$ has cardinality $d + 1 = 2$, there is only one non-∞ element s left that has to satisfy $s = 1$. ☐

For low degrees we can go a little further by determining interpolation sets leading to Boolean masking. Lemma 5 provides sufficient conditions for $d \leq 5$, we leave open the question for $d > 5$ though we suspect that for $\mathbb{F}_{2^8}$ equivalent polynomial schemes exist for most reasonable values of d.

Lemma 5. *Consider d-order Boolean masking over $\mathbb{F}_{2^e}$, then an interpolation set $\mathcal{S}$ for an equivalent polynomial masking scheme exists if $d = 1$, $d = 2$ and $e \geq 2$, $d = 3$ and $2|e$, $d = 4$ and $4|e$, or $d = 5$ and $5|e$.*

Proof. Below we state necessary and sufficient conditions on $\mathcal{S}$ for the four different values of d and establish how to instantiate:

$d = 1$: Lemma 4 already identified $\mathcal{S} = \{1, \infty\}$.

$d = 2$: As d is even, Lemma 4 implies that necessarily $\mathcal{S} = \{a, b, a+b\}$ for distinct, non-zero a and b. For sufficiency, we need that the final, third row sums to zero as well, but as $\sum_{s\in\mathcal{S}} s^2 = (\sum_{s\in\mathcal{S}} s)^2$ this is guaranteed for our choice of $\mathcal{S}$. Note that $e \geq 2$ is a necessary and sufficient condition for the existence of distinct non-zero a and b in $\mathbb{F}_{2^e}$.

$d = 3$: As d is odd and > 1, Lemma 4 implies that necessarily $\mathcal{S} = \{a, b, a+b, \infty\}$ for distinct, non-zero a and b. Again, $\sum_{s\in\mathcal{S}\setminus\{\infty\}} s^2 = (\sum_{s\in\mathcal{S}\setminus\{\infty\}} s)^2 = 0$, so the remaining necessary and sufficient condition on $\mathcal{S}$ is that $\sum_{s\in\mathcal{S}\setminus\{\infty\}} s^3 = 1$, or equivalently that $ab(a+b) + 1 = 0$; if $2|e$, then $\mathbb{F}_{2^e}$ contains a nontrivial third root of unity α and setting $a = 1, b = \alpha$ is a valid assignment as $1 \cdot \alpha(1+\alpha) + 1 = \alpha^2 + \alpha + 1 = 0$.

$d = 4$: Here Lemma 4 implies that $\mathcal{S} = \{a, b, c, d, a+b+c+d\}$ for distinct, non-zero a, b, c, d. As before, we automatically get that $\sum_{s\in\mathcal{S}} s^2 = 0$ and similarly we conclude that $\sum_{s\in\mathcal{S}} s^4 = 0$. Thus a necessary and sufficient condition on $\mathcal{S}$ is that $\sum_{s\in\mathcal{S}} s^3 = 0$. If $4|e$, then $\mathbb{F}_{2^e}$ contains a nontrivial fifth root of unity β and $\mathcal{S} = \{\beta, \beta^2, \beta^3, \beta^4, 1\}$ is a valid example as in that case cubing simply permutes the roots of unity, thus $\sum_{s\in\mathcal{S}} s^3 = \sum_{s\in\mathcal{S}} s = 0$ as desired.

$d = 5$: Here Lemma 4 implies that $\mathcal{S} = \{a, b, c, d, a + b + c + d, \infty\}$ for distinct, non-zero a, b, c, d. For $i \in \{2, \ldots, 4\}$ we require that $\sum_{s \in \mathcal{S} \setminus \{\infty\}} s^i = 0$ and additionally we require that $\sum_{s \in \mathcal{S} \setminus \{\infty\}} s^5 = 1$. As β from the $d = 4$ case is a fifth root of unity, $\beta^5 = 1$ and therefore setting $\mathcal{S} = \{\beta, \beta^2, \beta^3, \beta^4, 1, \infty\}$ works.

□

Having established that for suitably chosen interpolation sets, non-redundant polynomial masking is leak-equivalent to Boolean masking, a natural question is what happens when we consider redundant polynomial masking. Full leak-equivalence will be unlikely as Boolean masking is not redundant, but what happens when we take a "Boolean" interpolation set and add a point to it (increasing n), or evaluate the polynomial masking to a lower degree (reducing d)? In both cases, one might expect some of the Boolean behaviour still to be present, yet without being leak-equivalent to Boolean masking. We capture such behaviour under the moniker quasi-Boolean masking, as defined below.

Definition 3 (Quasi-Boolean Masking)**.** A code-based masking scheme is called quasi-Boolean of degree $d_\oplus$ when

$$d_\oplus = \left(\min_{\bar{\mathbf{d}} \in (\mathcal{C}^\perp \cap \{1\} \times \mathbb{F}_2^n)} \mathrm{Hw}(\bar{\mathbf{d}}) \right) - 2$$

is finite.

In other words, we are looking for a binary reconstruction vector $\mathbf{d}$ of the smallest weight possible, where $\bar{\mathbf{d}} = (1\mathbf{d})$. If no such $\mathbf{d}$ exists, then the scheme is not quasi-Boolean, otherwise $\mathrm{Hw}(\mathbf{d})$ shares suffice to reconstruct by simply adding those shares, corresponding to Boolean degree $\mathrm{Hw}(\mathbf{d}) - 1$. By definition, $d_\oplus \geq \min_{\bar{\mathbf{d}} \in \mathcal{C}^\perp \setminus \{0\}} \mathrm{Hw}(\bar{\mathbf{d}}) - 2$, i.e. the minimum distance of the dual code minus two. As we insisted on MDS codes (Definition 1), this minimum distance equals $d + 2$, thus $d_\oplus \geq d$.

Lemma 6 (Quasi-Booleanness for Small d)**.** *Let a $(1, n)$ polynomial masking scheme over $\mathbb{F}_{2^e}$ be given with interpolation set $\mathcal{S} \not\ni \infty$ and of quasi-Boolean degree $d_\oplus$. Then if $n > 2$, the $(2, n)$ polynomial masking scheme over $\mathbb{F}_{2^e}$ with the same interpolation set $\mathcal{S}$ is also quasi-Boolean with degree $d_\oplus$.*

Proof. Let $\bar{\mathbf{d}} = (1\mathbf{d})$ be an argument for which the minimum defining $d_\oplus$ is attained, so $\mathrm{Hw}(\mathbf{d}) = d_\oplus + 1$. Let $\mathcal{S}_\oplus$ consists of those elements in $\mathcal{S}$ where $\mathbf{d}$ is set. Then quasi-Booleanness implies that $\sum_{s \in \mathcal{S}_\oplus} s = 0$. As we are working in a characteristic 2 field, this implies that $\sum_{s \in \mathcal{S}_\oplus} s^2 = 0$ as well. □

3.4 Frobenius-stable Interpolation Sets

Roche and Prouff [RP12] suggest the use of interpolation sets that are stable under the fields Frobenius automorphism. For $\mathbb{F}_{2^8}$ they demonstrate existence of suitable sets (up to cardinality 255) by a not entirely constructive counting argument. For small cardinalities (up to and including 3) the stable set turns out to be unique when excluding 0 (and ignoring the point at infinity): for cardinality 1, only the identity suffices, and for cardinality 3 it is given as $\{1, \mathrm{0xbc}, \mathrm{0xbd}\}$. Remarkably, these points satisfy our criterion for quasi-Booleanness of degree $d_\oplus = 2$, even though the masking scheme itself only has degree $d = 1$. A natural question is to what extent other Frobenius stable interpolation sets are quasi-Boolean. Below we enumerate the possible Frobenius-stable sets for $n \leq 7$ and investigate their quasi-Booleanness.

For quasi-Booleanness of these sets, we are interested in odd-cardinality subsets $\mathcal{S}_\oplus$ of $\mathcal{S}$ such that $\sum_{s \in \mathcal{S}_\oplus} s^i = 0$ for $i = 1$, but ideally also for all $i > 0$ up to some maximum

Table 1: Frobenius-stable sets of interpolation points and their quasi-Booleanness.

n	$\mathcal{S}$	quasi-Boolean subsets for $(d_\oplus, d_{\max})$
2	$\{\alpha, \alpha+1\}$	safe
3	$\{1, \alpha, \alpha+1\}$	$\mathcal{S}_\oplus = \mathcal{S}_3$ for $(2,2)$
4a	$\{\beta, \beta^2, \beta^3, \beta^4\}$	safe
4b	$\{\beta+1, \beta^2+1, \beta^3+1, \beta^4+1\}$	safe
4c	$\{\beta+\beta^2, \beta+\beta^3, \beta^2+\beta^3, \beta^3+\beta^4\}$	safe
5a	$\mathcal{S}_{4a} \cup \{1\}$	$\mathcal{S}_\oplus = \mathcal{S}_{5a}$ for $(4,4)$
5b	$\mathcal{S}_{4b} \cup \{1\}$	$\mathcal{S}_\oplus = \mathcal{S}_{5b}$ for $(4,4)$
5c	$\mathcal{S}_{4c} \cup \{1\}$	$\mathcal{S}_\oplus = \mathcal{S}_{5c}$ for $(4,4)$
6a	$\mathcal{S}_{4a} \cup \mathcal{S}_2$	$\mathcal{S}_\oplus = \{\alpha, \beta, \beta^4\}$ for $(2,2)$
6b	$\mathcal{S}_{4b} \cup \mathcal{S}_2$	$\mathcal{S}_\oplus = \{\alpha, \beta+1, \beta^4+1\}$ for $(2,2)$
6c	$\mathcal{S}_{4c} \cup \mathcal{S}_2$	$\mathcal{S}_\oplus = \{\alpha, \beta+\beta^3, \beta^4+\beta^3\}$ for $(2,2)$
7a	$\mathcal{S}_{4a} \cup \mathcal{S}_3$	ex. $\mathcal{S}_\oplus = \mathcal{S}_3$ for $(2,2)$, $\mathcal{S}_\oplus = \mathcal{S}_{5a}$ for $(4,4)$
7b	$\mathcal{S}_{4b} \cup \mathcal{S}_3$	ex. $\mathcal{S}_\oplus = \mathcal{S}_3$ for $(2,2)$, $\mathcal{S}_\oplus = \mathcal{S}_{5b}$ for $(4,4)$
7c	$\mathcal{S}_{4c} \cup \mathcal{S}_3$	ex. $\mathcal{S}_\oplus = \mathcal{S}_3$ for $(2,2)$, $\mathcal{S}_\oplus = \mathcal{S}_{5c}$ for $(4,4)$

value $d_{\max}$. The cardinality of $\mathcal{S}_\oplus$ minus 1 indicates the quasi-Boolean degree $d_\oplus$, with $d_{\max}$ the highest degree d (of the masking) for which the quasi-Booleanness holds.

Our analysis works for any binary field whose extension degree is a multiple of 4, as it turns out that the relevant Frobenius-stable sets all live in $\mathbb{F}_{2^4}$. Recall that the multiplicative group of the finite field $\mathbb{F}_{2^4}$ is cyclic of order $15 = 3 \cdot 5$ and there exist unique subgroups of order 3 and 5, respectively.

The group of order 3 consists of $\{1, \alpha, \alpha^2\}$ where α is a root of the third cyclotomic polynomial, thus $\alpha^2 + \alpha + 1 = 0$ or $\alpha^2 = \alpha + 1$. Moreover, $(\alpha^2)^2 = \alpha$, thus $\mathcal{S}_2 = \{\alpha, \alpha+1\}$ is the unique Frobenius-stable set of cardinality 2 and $\mathcal{S}_3 = \{1, \alpha, \alpha+1\}$ is the unique Frobenius-stable set of cardinality 3 (and it matches the set explicitly given by Roche and Prouff for this size). For $\mathcal{S}_2$, there is no quasi-Boolean behaviour possible, but for $\mathcal{S}_3$, we can consider $\mathcal{S}_\oplus = \mathcal{S}_3$ and observe that $1^i + \alpha^i + (\alpha+1)^i$ equals 0 for $i \in \{1, 2\}$, but no longer for $i = 3$.

The group of order 5 consists of $\{1, \beta, \beta^2, \beta^3, \beta^4\}$ where β is a root of the fifth cyclotomic polynomial, thus those five elements sum to zero. Moreover, we can choose α and β such that $\alpha = \beta + \beta^4$ and $\alpha^2 = \beta^2 + \beta^3$. Then the three elements $\beta, \alpha\beta, \alpha^2\beta$ each generate (disjoint) Frobenius-stable sets of cardinality 4. Each cardinality 4 set can uniquely be extended to a cardinality 5 set by appending $\{1\}$, to a cardinality 6 set by appending $\{\alpha, \alpha^2\}$, or to a cardinality 7 set by appending $\{\alpha, \alpha^2, 1\}$. For the cardinality 4 set, no quasi-Boolean weakness exists, but for larger sets, more and more problems manifest itself.

Our results are summarized in Table 1. For instance, for row '5a', we see that $(d_\oplus, d_{\max}) = (4, 4)$, implying that using that set $\mathcal{S}_{4a} \cup \{1\}$ for polynomial masking of degree $d \leq d_{\max} = 4$ will result in quasi-Booleanness of degree $d_\oplus = 4$. We observe that opting for Frobenius-stable interpolation sets for any value other than $n \in \{2, 4\}$ introduces quasi-Booleanness up to second order masking schemes, so whenever $d \leq 2$. Consequently, Frobenius-stable interpolation sets are likely less secure than some generic interpolation set of the same size, at least when considering noisy Hamming weight leakage with the same noise level. For $d = 1$ we investigate later on (Section 5.1) and confirm that $\mathcal{S}_3$ is performing particularly poor, whereas $\mathcal{S}_4$ appears fine (for $n > 4$ we did not run any experiments).

Extending our analysis to $n \geq 8$ becomes slightly more tedious. For a Frobenius-stable set of cardinality 8, one could of course join any two distinct cardinality 4 sets (with quasi-Boolean degree 2 for either of the three resulting sets). For $\mathbb{F}_{2^4}$ this would more or less be the end of the story: for $8 \leq n \leq 11$ there are three possible Frobenius-stable sets each, whereas for $12 \leq n \leq 15$ the Frobenius-stable sets are unique. Moreover, as these larger Frobenius-stable sets are constructed from the smaller ones, they inherit and often amplify quasi-Boolean behaviour from below. For $\mathbb{F}_{2^8}$, the number of possible Frobenius-stable subsets of cardinality 8 is a lot higher. Indeed, Roche and Prouff indicated there to be 30, far exceeding our appetite for enumeration.

3.5 Polynomial Masking

Stepping away from special cases of polynomial masking, we briefly consider the more general case. Our main result is that it should be fairly safe to always include 1 in the evaluation set; the only real exception occurs when ∞ is around as well.

Lemma 7. *Let a (d, n) polynomial masking set with $\mathcal{S} \not\ni \infty$ be given, then there exists a leak-equivalent polynomial masking scheme with $1 \in \mathcal{S}'$.*

Proof. Let α_1 be the first element of $\mathcal{S} = \{\alpha_1, \ldots, \alpha_n\}$ used to create G, so we know $\alpha_1 \in \mathbb{F}^*$. Then the $(d+1) \times (d+1)$ diagonal matrix with α^i on its diagonal for $i = 0, \ldots, d$ provides the same interpolation code based on $\mathcal{S}' = \mathcal{S}/\alpha_1$. $\square$

In the special case of first order polynomial masking, there is a cute little corollary that an interpolation point and its inverse leak the same (both in conjuction with 1).

Corollary 1. *Let $\alpha \in \mathbb{F}^*$, then the $(1, 2)$ polynomial masking schemes defined by $\{1, \alpha\}$ and $\{1, \alpha^{-1}\}$ are leak-equivalent.*

Proof. By Lemma 3 interpolation based on $\{1, \alpha\}$ or on $\{\alpha, 1\}$ is fully leak-equivalent. Applying the procedure from the proof of Lemma 7 then produces $\{1, \alpha^{-1}\}$. $\square$

3.6 Revisited Inner Product Masking, Revisited

Balasch et al. [BFGV12] already described how to cast non-redundant polynomial masking as a special case of the later RIP [BFG15]. Indeed, RIP is presented as a generalization of both polynomial masking and of Boolean masking, we already saw how to mimic Boolean masking by polynomial masking using specially chosen interpolation sets $\mathcal{S}$. A natural question is to what extent we can mimic RIP as well by chosing suitable $\mathcal{S}$.

Previously, we cast d-order RIP as a code-based masking scheme based on a $[d+2, d+1]$ MDS code. Moreover, any $[d+2, d+1]$ MDS code—can be cast as a GRS code as long as $d+1 \leq |\mathbb{F}|$. The upper bound on d is a simple consequence of a field $\mathbb{F}$ of size q only providing q possible interpolation points (namely all elements of $\mathbb{F}^*$ plus the point at infinity).

The question however is whether we can use the more restrictive (shortened) Reed–Solomon codes (corresponding to polynomial masking) to capture RIP. We first provide a negative result (Lemma 8), by providing a much sharper upper bound on d for which full equivalence might be possible. For instance, if $q = 2^8$, then $d > 32$ forces inequivalence.

Lemma 8. *For any finite field $\mathbb{F}$ of size q, d-order RIP masking is more general than $(d, d+1)$ polynomial masking if $(q-1)^d/d! > \binom{q}{d+1}$.*

Proof. For a polynomial masking scheme, we get to select n distinct elements from a set of size $q = |\mathbb{F}|$. There are $\binom{q}{n}$ ways of selecting the set of interpolation points and then $n!$ ways of assigning the points to the parties. These $n!$ assignments evidently lead to equivalent codes, but it's even possible that different sets of interpolation points lead to equivalent or even identical codes (as we will see in a moment). Thus the number of non-equivalent codes using polynomial masking is at most $\binom{q}{n}$.

On the other hand, for RIP, we need to select $n-1$ coefficients from $\mathbb{F}^*$, where we allow duplicates. Thus there are $(q-1)^{n-1}$ ways of selecting these coefficients. Some of these selections will be equivalent as we can freely permute the coefficients to arrive at an equivalent code. Yet, for any given coefficient selection, there are at most $(n-1)!$ ways of permuting the coefficient, so there are at least $(q-1)^{n-1}/(n-1)!$

For a fixed q and sufficiently large n, we have that $(q-1)^{n-1}/(n-1)! > \binom{q}{n}$, indicating that RIP includes strictly more equivalence classes than polynomial masking. □

An obvious follow up question is whether there is any significant security benefit of RIP over polynomial masking for smaller, and arguably more realistic, masking order d. For instance, for $d = 1$ using polynomial masking with $\mathcal{S} = \{\lambda, \infty\}$ the reconstruction formula will be $(1\ \lambda)$, corresponding to RIP.

For larger d, we can establish a generalization of Lemma 4, as we do below.

Lemma 9. *Consider d-order RIP masking over $\mathbb{F}_{2^e}$ with reconstruction vector $\mathbf{d} \in (\mathbb{F}^*)^n$ subject to $d_n = 1$ and let $\mathcal{S}$ be an interpolation set for a leak-equivalent polynomial masking scheme. Then $\infty \in \mathcal{S}$ iff $\sum_{i=1}^{d} d_i = 1$.*

4 Maximum Likelihood Distinguisher

What it computes. When trying to solve the key recovery game (Figure 2), the optimal strategy for an adversary would be to output the key k that maximizes

$$\Pr[K = k \mid (L_1, \ldots, L_N) = (\ell_1, \ldots, \ell_N)] \ .$$

This so-called MAP estimate will equal the maximum likelihood estimate (MLE) if the key's prior is uniform, which it is. The MLE outputs the key that maximizes $p((L_1, \ldots, L_N) = (\ell_1, \ldots, \ell_N) \mid K = k)$. Taking logarithms and exploiting independence of the leakage (i.e. the invocations of Leak in the game), an adversary wants the key that minimizes $\sum_{i=1}^{N} \lg p(L_i = l_i \mid K = k)$. Moreover, $p(L_i = l_i \mid K = k) = \Pr[X_i = x_i] \cdot p(T_i = \mathbf{t}_i \mid K = k, X_i = x_i)$.

The first factor is irrelevant, so we can stick to $p(T_i = \mathbf{t}_i \mid K = k, X_i = x_i)$. In our setting, k and x_i uniquely determine the sensitive variable v_i that is leaked upon, so for a given v_i and $\mathbf{t}_i$ we are interested in $p_{\mathsf{Trace}}(\mathsf{Trace}(v_i) = \mathbf{t}_i)$, where the randomness is over the choices of the Trace code, which incorporates both the masking and the subsequent leakage on said masking.

We can expand the evaluation of Trace's pdf by making the masking explicit and exploiting that we drip on each share independently. Thus

$$p_{\mathsf{Trace}}(\mathsf{Trace}(v_i) = \mathbf{t}_i) = \sum_{\mathbf{c} \in \mathcal{C}(v_i)} \Pr_{\mathsf{mask}}[\mathsf{mask}(v_i) = \mathbf{c}] \cdot \prod_{j=1}^{n} p_{\mathsf{Drip}}(\mathsf{Drip}(c_j) = t_{ij})$$

where the factor $\Pr_{\mathsf{mask}}[\mathsf{mask}(v_i) = \mathbf{c}]$ represents a uniform choice which can safely be ignored.

We can summarize the discussion above by describing the distinguishing score $S(k)$ in terms of the given traces $\mathbf{t}_i$ and the k-dependent intermediate variables $v_i = \mathsf{S}(k \oplus x_i)$:

$$S(k) = \sum_{i=1}^{N} \lg \sum_{\mathbf{c} \in \mathcal{C}(v_i)} \prod_{j=1}^{n} p_{\mathsf{Drip}}(\mathsf{Drip}(c_j) = t_{ij}) = \sum_{i=1}^{N} \lg s(v_i, \mathbf{t}_i) \tag{1}$$

We stress that both MAP and MLE are very well-trodden concepts from machine learning and their application to side-channel analysis is well-known. The derivation above largely follows that of the "Higher-Order Optimal Distinguisher" [BGHR14, Theorems 2 and 7], adapted to our notation and setting that allows for redundant masking, with the further simplification of taking logarithms and ignoring $\Pr_{\mathsf{mask}}[\mathsf{mask}(v_i) = \mathbf{c}]$.

How to compute it. For the code-based masking schemes with noisy Hamming-weight leakage, the right hand side of 1 can be simplified further. Henceforth, we concentrate on the second summation only, namely the one captured by $s(v, \mathbf{t})$. Here we dropped the dummy variable 'i' so we can repurpose it in a moment.

Recall that for code-based masking of sensitive variable v, we select a random $\mathbf{u} \in \mathbb{F}_{2^e}^d$ and then calculate $\bar{\mathbf{c}} \leftarrow (v \ \mathbf{u}) \cdot G$. If G is the (transpose) Vandermonde matrix corresponding to polynomial masking with interpolation set $\mathcal{S} = \{\alpha_1, \ldots, \alpha_n\} \not\ni \infty$, then we can alternatively write

$$\forall_{1 \leq j \leq n} \ c_j \leftarrow v + \sum_{i=1}^{d} u_i \alpha_j^i \ .$$

For noisy Hamming-weight leakage,

$$p_{\mathsf{Drip}}\left(\mathsf{Drip}(c_j) = t_j\right) = \mathcal{N}\left(t_j \,\middle|\, \mathrm{Hw}(c_j), \sigma^2\right) \ ,$$

leading to the following

$$s(v, \mathbf{t}) = \sum_{\mathbf{u} \in \mathbb{F}_{2^e}^d} \prod_{j=1}^{n} \mathcal{N}\left(t_j \,\middle|\, \mathrm{Hw}(c_j), \sigma^2\right)$$

with c_j as described above and $\mathcal{N}\left(t_j \,\middle|\, \mathrm{Hw}(c_j), \sigma^2\right)$ describes the pdf of a Gaussian (normal) distribution of mean $\mathrm{Hw}(c_j)$ and variance σ^2 evaluated at t_j.

How not to compute it. Chabanne et al. [CMP18] based their distinguisher on an earlier adaptation of the likelihood distinguisher to Boolean masking [LPR+14, Eq. (32)]. For Boolean masking, one can evaluate (correctly)

$$s(v, \mathbf{t}) = \sum_{(c_1, \ldots, c_d) \in \mathbb{F}_{2^e}^d} \prod_{j=1}^{d+1} \mathcal{N}\left(t_j \,\middle|\, \mathrm{Hw}(c_j), \sigma^2\right)$$

where $c_{d+1} = \sum_{i=1}^{d} c_i$. A similar, reconstruction-centric distinguisher is possible for RIP, and was also suggested by Chabanne et al. for redundant masking, namely by evaluating

$$s(v, \mathbf{t}) = \sum_{(c_2, \ldots, c_n) \in \mathbb{F}_{2^e}^{n-1}} \prod_{j=1}^{n} \mathcal{N}\left(t_j \,\middle|\, \mathrm{Hw}(c_j), \sigma^2\right)$$

where $c_1 = \frac{1}{\beta_1}(v + \sum_{i=2}^{n} c_i \beta_i)$ and $\beta_i = \prod_{j=1, j \neq i}^{n} \frac{\alpha_j}{\alpha_i - \alpha_j}$. Essentially, the first share is recomputed based on reconstruction using all other remaining shares. (The reconstruction vector itself might not be entirely correct either, as there should be a dependency on the reconstruction set.)

Essentially, the Chabanne et al. distinguisher is still operating under the assumption that $n-1$ of the shares can be seen as independent random variables following a uniform distribution. However, when moving to redundant masking, this assumption is no longer true: the dimensions (or degrees of freedom) simply no longer match! It should always be d, not $n-1$, with the latter only being correct if it happens to match d (which is of course precisely the non-redundant case).

5 Measuring the Effect of Redundancy

All our experiment are conducted using the AES field-representation. We do not study the effect of changing the field representation in this work. The source code used for running the simulations and the experimental data with details of the experimental protocol can be found at `https://github.com/Simula-UiB/Redundant-Code-based-Masking`.

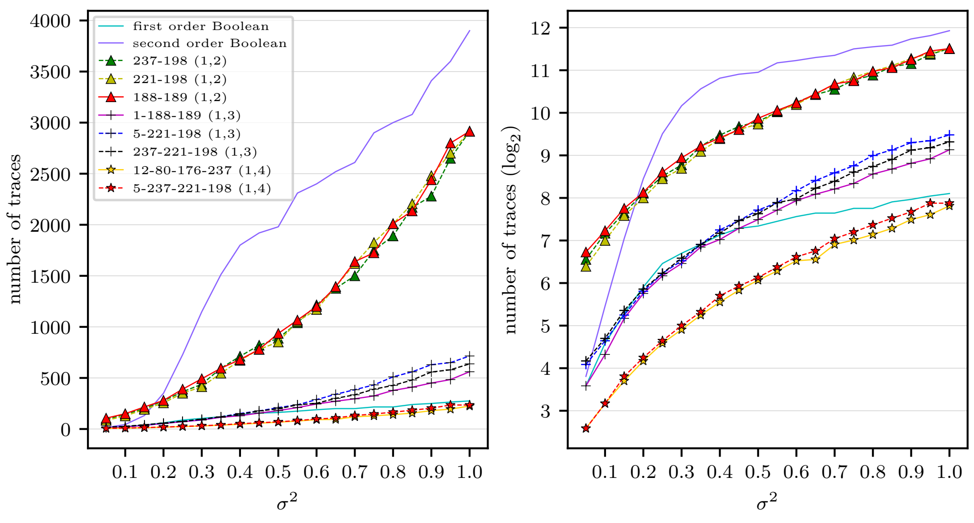

Figure 3: Attacks on $(1, n)$ masking scheme with sets of interpolation points stable under the Frobenius automorphism for $0.05 \leq \sigma^2 \leq 1.0$ and comparison with Boolean masking and "normal" sets of interpolation points.

5.1 Frobenius and Quasi-Boolean Polynomial Masking

We start with determining the quantitative effect of using interpolation sets $\mathcal{S}$ that are stable under the Frobenius automorphism. In Section 3.4 (Table 1) we investigated for which n we expect quasi-Boolean behaviour: $n = 2$ and $n = 4$ are not quasi-Boolean and should therefore perform relatively normal, whereas $n = 3$ is quasi-Boolean, which should yield much weaker security.

We ran attacks on $(1, n)$ polynomial masking with Frobenius stable sets for $2 \leq n \leq 4$ for $0.05 \leq \sigma^2 \leq 1.0$ (Figure 3) and $0.25 \leq \sigma^2 \leq 4.0$ (Figure 4). Given our leakage model, this range of σ^2 means the SNR ranges from 40 down to 1/2. The Frobenius sets are $\{188, 189\}$, $\{1, 188, 189\}$ and $\{12, 80, 176, 237\}$, respectively. For comparison and benchmarking, we also included first and second order Boolean masking, as well as two representative sets for $n = 3$ (the best and worst performing for very low noise). For $0.05 \leq \sigma^2 \leq 1.0$ are also included representative sets for $n = 2$ and $n = 4$.

We make several observations. Firstly, for the Frobenius sets that are not quasi-Boolean, so $n = 2$ and $n = 4$, there is no noticeable difference compared to the representative sets, confirming our hypothesis from Section 3.4 that these Frobenius stable sets do not lead to any security degradation. For this reason, we excluded them in Figure 4. Secondly, for $n = 3$ we observe very atypical and degraded behaviour. Initially it leaks even more than either representative set, but surprisingly it also out-leaks first-order Boolean masking. Somewhere between $0.25 \leq \sigma^2 \leq 0.4$ it catches up with both first-order Boolean and the representative sets. For $\sigma^2 > 0.4$ it starts to leak a lot worse again than the representative sets. At $\sigma^2 = 4$ the amount of traces is almost twice as low as for the worst performing representative set. This strongly points towards quasi-Boolean yielding a much weaker security in general. An interesting question that remains is whether for even higher σ^2 the $(1, 3)$ Frobenius stable set eventually catches up on second order Boolean masking.

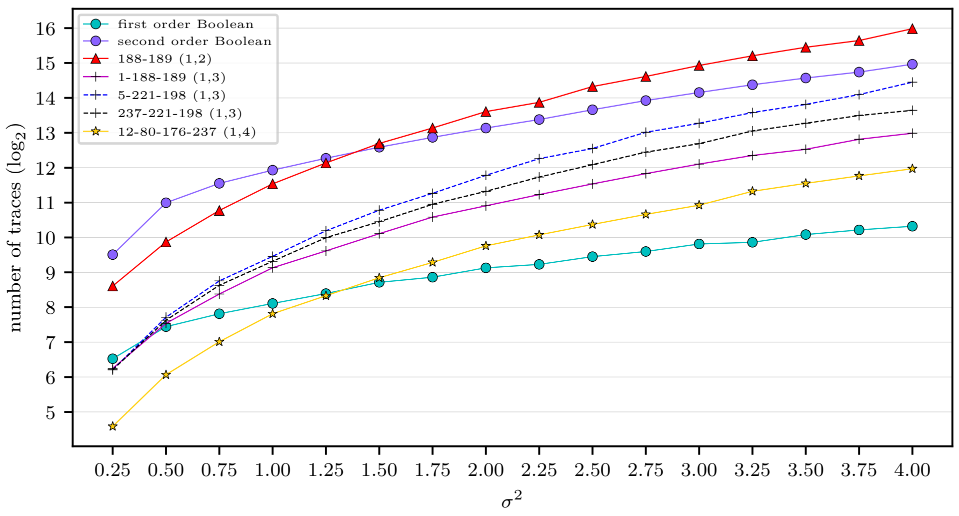

Figure 4: Attacks on $(1, n)$ with sets of interpolation points stable under the Frobenius automorphism for $0.25 \leq \sigma^2 \leq 4.0$ and comparison with Boolean masking and "normal" sets of interpolation points.

Boolean. As explained in Section 3, a $(2, 3)$ polynomial masking scheme with $\mathcal{S} = \{a, b, a + b\}$ is leak-equivalent to second-order Boolean masking. Formally, we could not prove *full* leak-equivalence, thus to gain extra insight, we run attacks on second order Boolean masking as well as two distinct, Boolean-like $(2, 3)$ polynomial masking schemes, using {1, 188, 189} and {1, 146, 147}, respectively. The results are shown in Figure 5.

The experiments demonstrate that the practice matches the theory, as both Boolean-like polynomial masking schemes closely track the second-order Boolean scheme. For a sufficiently low SNR and well-chosen evaluation points, first-order polynomial masking is more secure than second-order Boolean masking [RP12, Figure 3]. By extension, first-order polynomial masking can also end up being *more* secure than second order polynomial masking with a particularly poor selection of evaluation points, highlighting even more the crucial need of careful selection of interpolation points.

5.2 "Normal" Redundant Polynomial Masking

We now turn our attention to the effect of redundancy for the more general case. We use the set $\{5, 175, 198, 221, 237\}$, as previously used by Chabanne et al. [CMP18], as being "representative". We focus on $d = 1$ and range n from 2 up to 5, where for $n < 5$ we report on various possible subsets of the master set (our selection here was governed by the behaviour for very low noise).

The results, presented in Figure 6, clearly demonstrate the distinguishing gain by exploiting as many shares as available, rather than just targeting the leakiest subset of $d + 1$ points (as suggested by Chabanne et al. [CMP18] for some SNR). Regardless of the SNR, redundant shares can always be exploited by the distinguisher and the more redundant shares are available, the easier it gets to recover the key. Thus, when introducing redundancy to guard against other threats (glitches, faults), it is important to realize that the trade-off may not just be more security for less performance, but can actually require a balancing of various threat models, resulting in improved active security for reduced passive security.

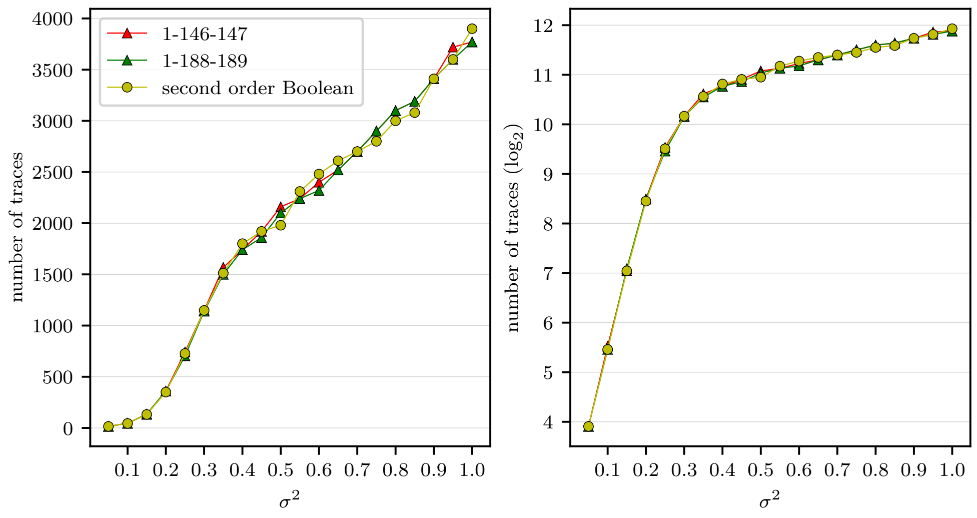

Figure 5: Attacks on "boolean case" (3,2) polynomial masking and second order masking.

When looking at the curves in lg scale, we observe that they are roughly translations of each other. Thus, the difference between the curves at $\sigma^2 = 0.05$ is a very appropriate approximation of the difference at $\sigma^2 = 1.0$. Thus, we can compactly represent the effect of redundancy concentrating on $\sigma^2 = 0.05$ and compare the various choices for n and d. We already reported on these results in the Introduction, using Figure 1. That Figure contains a few more data points, namely $n = 6$ for $d = 1$, as well as $2 \leq n \leq 5$ for $(2, n)$. Nonetheless, the number of data points available is too small to fit a function in order to establish a suitable rule of thumb capturing the quantitative security degradation as a consequence of redundant masking.

In a practical setting, exploiting more shares almost certainly increases the cost of extracting features from real traces, more difficulty to align the traces properly and more involved and costly profiling. Moreover, against standard, non-redundant masking schemes it is well-known how to run non-profiled attacks, for instance by using leakage combiners [OM07, SVO$^+$10a, BGHR14]. From that perspective, the most practical attack in real-life might only exploit the minimal amount of shares available, or at least not all. Then again, an attacker with a lot of processing power but only limited access to traces, might want to try to exploit redundant shares as much as possible (cf. the application of belief propagation to side-channel attacks [VGS14, GRO18]).

Finally, in the Introduction we suggested that one could turn a single trace leaking on n shares into $\binom{n}{d+1}$ tuples of traces, each leaking on $d + 1$ shares. Against each tuple, one could then run an unprofiled attack against the suspected $d + 1$ traces (e.g. by using a combiner) and then merge the resulting scores. This could lead to a substantial acceleration to recover the secret in a potentially practical way.

6 Conclusion

We investigated polynomial masking through the prism of code-based masking, allowing us to consider leak-equivalence when comparing different classes of masking schemes. Code-based masking scheme is properly parameterized by its generator matrix G and classical schemes such as polynomial masking, Boolean masking, and RIP all impose structure by

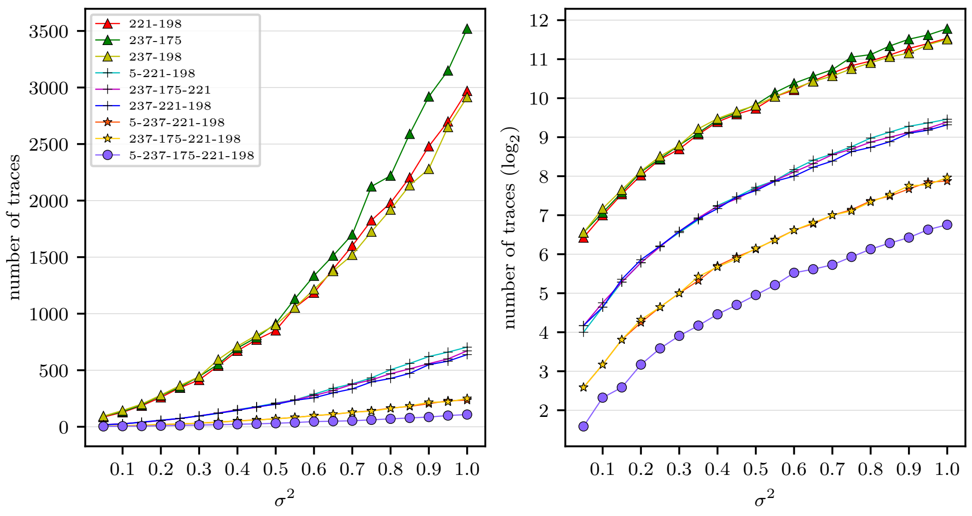

Figure 6: Attack on redundant masking scheme with $0.05 \leq \sigma^2 \leq 1.0$ and $2 \leq n \leq 5$.

considering only a subset of the full parameter space. On the one hand, the additional structure might speed up computations by allowing for specialized gadgets, but on the other hand, the more general schemes allow a larger search space to explore in order to minimize leakage.

When considering noisy Hamming weight leakage on individual shares of a typical sensitive variable only, the security loss of a more specialized parameter selection can be significant, as we saw for the Frobenius-stable polynomial evaluation, which we identified as quasi-Boolean for $n = 3$. For real implementations, the specialized gadgets tend to be faster, so perhaps one should expect them to leak less (lower SNR), potentially offsetting the higher leakage for identical SNR. We leave open this fascinating possibility.

If we fix d and σ^2, then security in our noisy model decreases for increasing n. This holds for any σ^2 under investigation, contrary to the claim by Chabanne et al. [CMP18] that for larger σ^2 (within our range) it might be advantageous to ignore some leakages. We also challenge the claim by Seker et al. [SFRES18] that $(1, 4)$, $(1, 5)$, and $(1, 6)$ polynomial masking (using Frobenius stable interpolation sets) all have the same side-channel resistance: certainly in the noisy Hamming weight model one has to expect serious security degradation (cf. Table 1). Our results do chime with an observation by Bruneau et al. [BGHR14, Section 6] in the setting of attacking a typical masked AES evaluation pattern based on first order Boolean masking. They write “The correct answer is to prevent from selecting only two leakages when more are available, and instead exploit all of them simultaneously in a single attack.” We concur.

Finally, n has no bearing on the d-probing security of the sharing. This contrasts with the result [DFS19, Corollary 1] that for dth order Boolean masking, any $0 < \gamma < 1$ and sufficiently large σ, the key recovery advantage based on N traces can be bounded by $1 - N\gamma^d$. Although our result is compatible with that earlier result, there does appear to be some tension. It would be interesting to see how a concrete generalization of the probing-model-to-noisy-leakage result [DDF19] to more redundant code-based masking would pan out and where the necessary dependency on n comes into play. For instance, what entails “sufficiently large” could well depend on n.

References

[BBD$^+$16] Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini. Strong non-interference and type-directed higher-order masking. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 116–129. ACM Press, October 2016.

[BCC$^+$14] 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 *WISTP*, volume 8501 of *LNCS*, pages 40–56. Springer, Heidelberg, 2014.

[BCPZ16] Alberto Battistello, Jean-Sébastien Coron, Emmanuel Prouff, and Rina Zeitoun. Horizontal side-channel attacks and countermeasures on the ISW masking scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *CHES 2016*, volume 9813 of *LNCS*, pages 23–39. Springer, Heidelberg, August 2016.

[BFG15] Josep Balasch, Sebastian Faust, and Benedikt Gierlichs. Inner product masking revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 486–510. Springer, Heidelberg, April 2015.

[BFGV12] Josep Balasch, Sebastian Faust, Benedikt Gierlichs, and Ingrid Verbauwhede. Theory and practice of a leakage resilient masking scheme. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 758–775. Springer, Heidelberg, December 2012.

[BGHR14] Nicolas Bruneau, Sylvain Guilley, Annelie Heuser, and Olivier Rioul. Masks will fall off - higher-order optimal distinguishers. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part II*, volume 8874 of *LNCS*, pages 344–365. Springer, Heidelberg, December 2014.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. *Proceedings of AFIPS 1979 National Computer Conference*, 48:313–317, 1979.

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[CGC$^+$20] Wei Cheng, Sylvain Guilley, Claude Carlet, Sihem Mesnager, and Jean-Luc Danger. Optimizing inner product masking scheme by A coding theory approach. *IACR Cryptol. ePrint Arch.*, 2020:692, 2020.

[CJRR99] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 398–412. Springer, Heidelberg, August 1999.

[CMP18] Hervé Chabanne, Houssem Maghrebi, and Emmanuel Prouff. Linear repairing codes and side-channel attacks. *IACR TCHES*, 2018(1):118–141, 2018. https://tches.iacr.org/index.php/TCHES/article/view/835.

[CPR12] Jean-Sébastien Coron, Emmanuel Prouff, and Thomas Roche. On the use of Shamir's secret sharing against side-channel analysis. In *CARDIS*, volume 7771 of *LNCS*, pages 77–90. Springer, Heidelberg, 2012.

[CRZ13] Guilhem Castagnos, Soline Renner, and Gilles Zémor. High-order masking by using coding theory and its application to AES. In Martijn Stam, editor, *14th IMA International Conference on Cryptography and Coding*, volume 8308 of *LNCS*, pages 193–212. Springer, Heidelberg, December 2013.

[DDF19] Alexandre Duc, Stefan Dziembowski, and Sebastian Faust. Unifying leakage models: From probing attacks to noisy leakage. *Journal of Cryptology*, 32(1):151–177, January 2019.

[DF12] Stefan Dziembowski and Sebastian Faust. Leakage-resilient circuits without computational assumptions. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 230–247. Springer, Heidelberg, March 2012.

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

[dHVdV$^+$03] Jerry den Hartog, Jan Verschuren, Erik P. de Vink, Jaap de Vos, and W. Wiersma. PINPAS: A tool for power analysis of smartcards. In *SEC*, volume 250 of *IFIP Conference Proceedings*, pages 453–457. Kluwer, 2003.

[GM11] Louis Goubin and Ange Martinelli. Protecting AES with Shamir's secret sharing scheme. In Bart Preneel and Tsuyoshi Takagi, editors, *CHES 2011*, volume 6917 of *LNCS*, pages 79–94. Springer, Heidelberg, September / October 2011.

[GMPO19] Si Gao, Ben Marshall, Dan Page, and Elisabeth Oswald. Share-slicing: Friend or foe? *IACR TCHES*, 2020(1):152–174, 2019. https://tches.iacr.org/index.php/TCHES/article/view/8396.

[GRO18] Joey Green, Arnab Roy, and Elisabeth Oswald. A systematic study of the impact of graphical models on inference-based attacks on AES. In *CARDIS*, volume 11389 of *LNCS*, pages 18–34. Springer, Heidelberg, 2018.

[GSF14] Vincent Grosso, François-Xavier Standaert, and Sebastian Faust. Masking vs. multiparty computation: how large is the gap for AES? *Journal of Cryptographic Engineering*, 4(1):47–57, April 2014.

[HRG14] Annelie Heuser, Olivier Rioul, and Sylvain Guilley. Good is not good enough - deriving optimal distinguishers from communication theory. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 55–74. Springer, Heidelberg, September 2014.

[ISW03] Yuval Ishai, Amit Sahai, and David Wagner. Private circuits: Securing hardware against probing attacks. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 463–481. Springer, Heidelberg, August 2003.

[KPP20] Matthias J. Kannwischer, Peter Pessl, and Robert Primas. Single-trace attacks on Keccak. *IACR TCHES*, 2020(3):243–268, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8590.

[LPR+14] Victor Lomné, Emmanuel Prouff, Matthieu Rivain, Thomas Roche, and Adrian Thillard. How to estimate the success rate of higher-order side-channel attacks. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 35–54. Springer, Heidelberg, September 2014.

[Mas93] James L. Massey. Minimal codewords and secret sharing. In *Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory*, pages 276–279, 1993.

[MM13] Amir Moradi and Oliver Mischke. On the simplicity of converting leakages from multivariate to univariate - (case study of a glitch-resistant masking scheme). In Guido Bertoni and Jean-Sébastien Coron, editors, *CHES 2013*, volume 8086 of *LNCS*, pages 1–20. Springer, Heidelberg, August 2013.

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

[MOW17] David McCann, Elisabeth Oswald, and Carolyn Whitnall. Towards practical tools for side channel aware software engineering: 'grey box' modelling for instruction leakages. In Engin Kirda and Thomas Ristenpart, editors, *USENIX Security 2017*, pages 199–216. USENIX Association, August 2017.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, 1977.

[MS81] Robert J. McEliece and Dilip V. Sarwate. On sharing secrets and reed-solomon codes. *Commun. ACM*, 24(9):583–584, 1981.

[OM07] Elisabeth Oswald and Stefan Mangard. Template attacks on masking - resistance is futile. In Masayuki Abe, editor, *CT-RSA 2007*, volume 4377 of *LNCS*, pages 243–256. Springer, Heidelberg, February 2007.

[PGS+17] Romain Poussier, Qian Guo, François-Xavier Standaert, Claude Carlet, and Sylvain Guilley. Connecting and improving direct sum masking and inner product masking. In *CARDIS*, volume 10728 of *LNCS*, pages 123–141. Springer, Heidelberg, 2017.

[PR11] 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 2011*, volume 6917 of *LNCS*, pages 63–78. Springer, Heidelberg, September / October 2011.

[PR13] Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 142–159. Springer, Heidelberg, May 2013.

[RP12] Thomas Roche and Emmanuel Prouff. Higher-order glitch free implementation of the AES using secure multi-party computation protocols - extended version. *Journal of Cryptographic Engineering*, 2(2):111–127, September 2012.

[SFRES18] Okan Seker, Abraham Fernandez-Rubio, Thomas Eisenbarth, and Rainer Steinwandt. Extending glitch-free multiparty protocols to resist fault injection attacks. *IACR TCHES*, 2018(3):394–430, 2018. https://tches.iacr.org/index.php/TCHES/article/view/7281.

[Sha79] Adi Shamir. How to share a secret. *Communications of the Association for Computing Machinery*, 22(11):612–613, November 1979.

[SVO$^+$10a] François-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 112–129. Springer, Heidelberg, December 2010.

[SVO$^+$10b] Francois-Xavier Standaert, Nicolas Veyrat-Charvillon, Elisabeth Oswald, Benedikt Gierlichs, Marcel Medwed, Markus Kasper, and Stefan Mangard. The world is not enough: Another look on second-order DPA. Cryptology ePrint Archive, Report 2010/180, 2010. http://eprint.iacr.org/2010/180.

[TAL09] Céline Thuillet, Philippe Andouard, and Olivier Ly. A smart card power analysis simulator. In *CSE (2)*, pages 847–852. IEEE Computer Society, 2009.

[VGS14] Nicolas Veyrat-Charvillon, Benoît Gérard, and François-Xavier Standaert. Soft analytical side-channel attacks. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 282–296. Springer, Heidelberg, December 2014.

[vL99] Jacobus H. van Lint. *Introduction to Coding Theory (Third Edition)*, volume 86 of *Graduate Texts in Mathematics*. Springer, 1999.

[WMCS20] Weijia Wang, Pierrick Méaux, Gaëtan Cassiers, and François-Xavier Standaert. Efficient and private computations with code-based masking. *IACR TCHES*, 2020(2):128–171, 2020. https://tches.iacr.org/index.php/TCHES/article/view/8547.

[WSY$^+$16] Weijia Wang, François-Xavier Standaert, Yu Yu, Sihang Pu, Junrong Liu, Zheng Guo, and Dawu Gu. Inner product masking for bitslice ciphers and security order amplification for linear leakages. In *CARDIS*, volume 10146 of *LNCS*, pages 174–191. Springer, Heidelberg, 2016.