New Instantiations of the CRYPTO 2017 Masking Schemes

Pierre Karpman[1] and Daniel S. Roche[2]

[1] Univ. Grenoble Alpes, CNRS, Grenoble INP[†], LJK, 38000 Grenoble, France
[2] United States Naval Academy, U.S.A.

pierre.karpman@univ-grenoble-alpes.fr, roche@usna.edu

Abstract. At CRYPTO 2017, Belaïd *et al.* presented two new private multiplication algorithms over finite fields, to be used in secure masking schemes. To date, these algorithms have the lowest known complexity in terms of *bilinear* multiplication and random masks respectively, both being linear in the number of shares $d + 1$. Yet, a practical drawback of both algorithms is that their safe *instantiation* relies on finding matrices satisfying certain conditions. In their work, Belaïd *et al.* only address these up to $d = 2$ and 3 for the first and second algorithm respectively, limiting so far the practical usefulness of their constructions.
In this paper, we use in turn an algebraic, heuristic, and experimental approach to find many more safe instances of Belaïd *et al.*'s algorithms. This results in explicit instantiations up to order $d = 6$ over large fields, and up to $d = 4$ over practically relevant fields such as $\mathbb{F}_{2^8}$.

Keywords: Masking, linear algebra, MDS matrices.

1 Introduction

It has become a well-accepted fact that the black-box security of a cryptographic scheme and the security of one of its real-life implementations may be two quite different matters. In the latter case, numerous side-channels or fault injection techniques may be used to aid in the cryptanalysis of what could otherwise be a very sensible design (for instance a provably-secure mode of operation on top of a block cipher with no known dedicated attacks).

A successful line of side-channel attacks is based on the idea of differential power analysis (DPA), which was introduced by Kocher, Jaffe and Jun at CRYPTO'99 [KJJ99]. The practical importance of this threat immediately triggered an effort from cryptographers to find adequate protections. One of the notable resulting counter-measures is the *masking* approach from Chari *et al.* and Goubin & Patarin [CJRR99,GP99]. The central idea of this counter-measure is to add a "mask" to sensitive variables whose observation through a side-channel

[†]Institute of Engineering Univ. Grenoble Alpes

could otherwise leak secret information; such variables are for instance intermediate values in a block cipher computation that depend on a known plaintext and a round key. Masking schemes apply a secret-sharing technique to several masked instances of every sensitive variable: a legitimate user knowing all the shares can easily compute the original value, while an adversary is now forced to observe more than one value in order to learn anything secret. The utility of this overall approach is that it is experimentally the case that the work required to observe n values accurately through DPA increases exponentially with n.

The challenge in masking countermeasures is to find efficient ways to compute with shared masked data while maintaining the property that the observation of n intermediate values is necessary to learn a secret (for some parameter n). When computations are specified as arithmetic circuits over a finite field $\mathbb{F}_q$, this task reduces mostly to the specification of secure shared addition and multiplication in that field. A simple and commonly used secret sharing scheme used in masking is the linear mapping $x \mapsto \left(r_1, \ldots, r_d, x + \sum_{i=1}^{d} r_i\right)$ which makes addition trivial; the problem then becomes how to *multiply* shared values. At CRYPTO 2003, Ishai, Sahai and Wagner introduced exactly such a shared multiplication over $\mathbb{F}_2$, proven secure in a *d-probing model* that they introduced [ISW03]. Their scheme requires $d(d+1)/2$ random field elements (*i.e.* bits) and $(d+1)^2$ field multiplications to protect against an adversary able to observe d intermediate values. This relatively high quadratic complexity in the *order* d of the scheme lead to an effort to decrease the theoretical and/or practical cost of masking.

At EUROCRYPT 2016, Belaïd *et al.* presented a masking scheme over $\mathbb{F}_2$ with *randomness complexity* decreased to $d + d^2/4$; implementations at low but practically relevant orders $d \leq 4$ confirmed the gain offered by their new algorithm [BBP+16]. At CRYPTO 2017, the same authors presented two new private multiplication algorithms over arbitrary finite fields [BBP+17]. The first, *Algorithm 4*, decreases the number of *bilinear* multiplications to $2d+1$ at the cost of additional constant multiplications and increased randomness complexity; the second, *Algorithm 5*, decreases the randomness complexity to only d, at the cost of $d(d+1)$ constant multiplications. Furthermore, both algorithms are proven secure w.r.t. the strong, composable notions of d-(strong) non-interference from Barthe *et al.* [BBD+16]. Yet a practical drawback of these last two algorithms is that their safe instantiation depends on finding matrices satisfying a certain number of conditions. Namely, Algorithm 4 uses two (related) matrices in $\mathbb{F}_q^{d \times d}$ for an instantiation at order $d+1$ over $\mathbb{F}_q$, while Algorithm 5 uses a single matrix in $\mathbb{F}_q^{d+1 \times d}$ for the same setting. In their paper, Belaïd *et al.* only succeed in providing "safe matrices" for the small cases $d = 2$ and $d = 2, 3$ for Algorithms 4 and 5 respectively, and in giving a non-constructive existence theorem for safe matrices when $q \geq \mathrm{O}(d)^{d+1}$ (resp. $q \geq \mathrm{O}(d)^{d+2}$).

1.1 Our contribution

In this work, we focus on the problem of safely instantiating the two algorithms of Belaïd *et al.* from CRYPTO 2017. We first develop equivalent matrix conditions

which are in some sense simpler and much more efficient to check computationally. We use this reformulation to develop useful *preconditions* based on MDS matrices that increase the likelihood that a given matrix is safe. We show how to generate matrices that satisfy our preconditions by construction, which then allows to give an explicit sufficient condition, as well as a construction of safe matrices for both schemes at order $d \leq 3$. Our simplification of the conditions also naturally transforms into a testing algorithm, an efficient implementation of which is used to perform an extensive experimental search. We provide explicit matrices for safe instantiations in all of the following cases:

- For $d = 3$, fields $\mathbb{F}_{2^k}$ with $k \geq 3$
- For $d = 4$, fields $\mathbb{F}_{2^k}$ with $5 \leq k \leq 16$
- For $d = 5$, fields $\mathbb{F}_{2^k}$ with $10 \leq k \leq 16$, and additionally $k = 9$ for Algorithm 5.
- For $d = 6$, fields $\mathbb{F}_{2^k}$ with $15 \leq k \leq 16$

These are the first known instantiations for $d \geq 4$ or for $d = 3$ over $\mathbb{F}_{2^3}$. We also gather detailed statistics about the proportion of safe matrices in all of these cases.

1.2 Roadmap

We recall the two masking schemes of CRYPTO 2017 and the associated matrix conditions in Section 3. We give our simplifications of the latter in Section 4 and state our preconditions in Section 5. A formal analysis of the case of order up to 3 is given in Section 6, where explicit conditions and instantiations for these orders are also developed. We present our algorithms and discuss their implementations in Section 7, and conclude with experimental results in Section 8.

2 Preliminaries

2.1 Notation

We use $\mathbb{K}^{m \times n}$ to denote the set of matrices with m rows and n columns over the field $\mathbb{K}$. We write $m = \operatorname{rowdim} A$ and $n = \operatorname{coldim} \boldsymbol{A}$. For any vector $\boldsymbol{v}$, $\operatorname{wt}(\boldsymbol{v})$ denotes the *Hamming weight* of $\boldsymbol{v}$, *i.e.*, the number of non-zero entries.

We use $\mathbf{0}_{m \times n}$ (resp. $\mathbf{1}_{m \times n}$) to denote the all-zero (resp. all-one) matrix in $\mathbb{K}^{m \times n}$ for any fixed $\mathbb{K}$ (which will always be clear from the context). Similarly, $\boldsymbol{I}_d$ is the identity matrix of dimension d.

We generally use bold upper-case to denote matrices and bold lower-case to denote vectors. (The exception is some lower-case Greek letters for matrices that have been already defined in the literature, notably $\boldsymbol{\gamma}$.) For a matrix $\boldsymbol{M}$, $\boldsymbol{M}_{i,j}$ is the coefficient at the ith row and jth column, with numbering (usually) starting from one. (Again, $\boldsymbol{\gamma}$ will be an exception as its row numbering starts at 0.) Similarly, a matrix may be directly defined from its coefficients as $\left(\boldsymbol{M}_{i,j}\right)$.

We use "hexadecimal notation" for binary field elements. This means that $a = \sum_{i=0}^{n-1} a_i X^i \in \mathbb{F}_{2^n} \cong \mathbb{F}_2[X]/\langle I(X) \rangle$ (where $I(X)$ is a degree-n irreducible

polynomial) is equated to the integer $\tilde{a} = \sum_{i=0}^{n-1} a_i 2^i$, which is then written in base 16. The specific field representations we use throughout are:

$\mathbb{F}_{2^2} \cong \mathbb{F}_2[x]/\langle X^2 + X + 1\rangle$	$\mathbb{F}_{2^3} \cong \mathbb{F}_2[x]/\langle X^3 + X + 1\rangle$
$\mathbb{F}_{2^4} \cong \mathbb{F}_2[x]/\langle X^4 + X + 1\rangle$	$\mathbb{F}_{2^5} \cong \mathbb{F}_2[x]/\langle X^5 + X^2 + 1\rangle$
$\mathbb{F}_{2^6} \cong \mathbb{F}_2[X]/\langle X^6 + X + 1\rangle$	$\mathbb{F}_{2^7} \cong \mathbb{F}_2[X]/\langle X^7 + X + 1\rangle$
$\mathbb{F}_{2^8} \cong \mathbb{F}_2[X]/\langle X^8 + X^4 + X^3 + X + 1\rangle$	$\mathbb{F}_{2^9} \cong \mathbb{F}_2[X]/\langle X^9 + X + 1\rangle$
$\mathbb{F}_{2^{10}} \cong \mathbb{F}_2[X]/\langle X^{10} + X^3 + 1\rangle$	$\mathbb{F}_{2^{11}} \cong \mathbb{F}_2[X]/\langle X^{11} + X^2 + 1\rangle$
$\mathbb{F}_{2^{12}} \cong \mathbb{F}_2[X]/\langle X^{12} + X^3 + 1\rangle$	$\mathbb{F}_{2^{13}} \cong \mathbb{F}_2[X]/\langle X^{13} + X^4 + X^3 + X + 1\rangle$
$\mathbb{F}_{2^{14}} \cong \mathbb{F}_2[X]/\langle X^{14} + X^5 + 1\rangle$	$\mathbb{F}_{2^{15}} \cong \mathbb{F}_2[X]/\langle X^{15} + X + 1\rangle$
$\mathbb{F}_{2^{16}} \cong \mathbb{F}_2[X]/\langle X^{16} + X^5 + X^3 + X + 1\rangle$	

Additional notation is introduced on first use.

2.2 MDS & Cauchy matrices

An $[n, k, d]_{\mathbb{K}}$ linear code of length n, dimension k, minimum distance d over the field $\mathbb{K}$ is *maximum-distance separable* (MDS) if it reaches the Singleton bound, *i.e.* if $d = n - k + 1$. An *MDS matrix* is the redundancy part $\boldsymbol{A}$ of a systematic generating matrix $\boldsymbol{G} = \begin{pmatrix}\boldsymbol{I}_k & \boldsymbol{A}\end{pmatrix}$ of a (linear) MDS code of length double its dimension.

A useful characterization of MDS matrices of particular interest in our case is stated in the following theorem (see *e.g.* [MS06, Chap. 11, Thm. 8]):

Theorem 1. *A matrix is MDS if and only if all its minors are non-zero,* i.e. *all its square sub-matrices are invertible.*

Square *Cauchy matrices* satisfy the above condition by construction, and are thence MDS. A (non-necessarily square) matrix $\boldsymbol{A} \in \mathbb{K}^{n\times m}$ is a Cauchy matrix if $\boldsymbol{A}_{i,j} = (x_i - y_j)^{-1}$, where $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$ are $n+m$ distinct elements of $\mathbb{K}$.

A Cauchy matrix $\boldsymbol{A}$ may be *extended* to a matrix $\widetilde{\boldsymbol{A}}$ by adding a row or a column of ones. It can be shown that all square submatrices of $\widetilde{\boldsymbol{A}}$ are invertible, and thus themselves MDS [RS85]. By analogy and by a slight abuse of terminology, we will say of a square matrix $\boldsymbol{A}$ that it is *extended MDS* (XMDS) if all square submatrices of $\boldsymbol{A}$ extended by one row or column of ones are MDS. Further depending on the context, we may only require this property to hold for row (or column) extension to call a matrix XMDS.

A (possibly extended) Cauchy matrix $\boldsymbol{A}$ may be *generalized* to a matrix $\boldsymbol{A'}$ by multiplying it with (non-zero) row and column scaling: one has $\boldsymbol{A'}_{i,j} = c_i d_j \cdot (x_i - y_j)^{-1}$, $c_i d_j \neq 0$. All square submatrices of generalized (extended) Cauchy matrices are MDS [RS85], but not necessarily XMDS, as one may already use the scaling to set any row or column of $\boldsymbol{A'}$ to an arbitrary value.

2.3 Security notions for masking schemes

We recall the security notions under which the masking schemes studied in this paper were analysed. These are namely *d-non-interference* (*d*-NI) and *d-strong*

non-interference (d-SNI), which were both introduced by Barthe *et al.* [BBD+16] as stronger and composable alternatives to the original d-probing model of Ishai *et al.* [ISW03].

Note that none of the notions presented below are explicitly used in this paper, and we only present them for the sake of completeness. Our exposition is strongly based on the one of Belaïd *et al.* [BBP+17].

Definition 2 (Gadgets). *Let $f : \mathbb{K}^n \to \mathbb{K}^m$, u, $v \in \mathbb{N}$; a (u, v)-*gadget *for the function f is a randomized circuit C such that for every tuple $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in (\mathbb{K}^u)^n$ and every set of random coins $\mathcal{R}$, $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m) \leftarrow\!\!\shortmid C(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n; \mathcal{R})$ satisfies:*

$$\left(\sum_{j=1}^{v} \boldsymbol{y}_{1,j}, \ldots, \sum_{j=1}^{v} \boldsymbol{y}_{m,j}\right) = f\left(\sum_{j=1}^{u} \boldsymbol{x}_{1,j}, \ldots, \sum_{j=1}^{u} \boldsymbol{x}_{m,j}\right).$$

One further defines x_i as $\sum_{j=1}^{u} \boldsymbol{x}_{i,j}$, and similarly for y_i; $\boldsymbol{x}_{i,j}$ is called the jth share *of x_i.*

In the above, the randomized circuit C has access to random-scalar gates that generate elements of $\mathbb{K}$ independently and uniformly at random, and the variable $\mathcal{R}$ records the generated values for a given execution. Furthermore, one calls *probes* any subset of the wires of C (or equivalently edges of its associated graph).

Definition 3 (t-Simulability). *Let C be a (u, v)-gadget for $f : \mathbb{K}^n \to \mathbb{K}^n$, and ℓ, $t \in \mathbb{N}$. A set $\{p_1, \ldots, p_\ell\}$ of probes of C is said to be* t-simulable *if $\exists\, I_1, \ldots, I_n \subseteq \{1, \ldots, u\}$; $\#I_i \leq t$ and a randomized function $\pi : (\mathbb{K}^t)^n \to \mathbb{K}^\ell$ such that for any $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in (\mathbb{K}^u)^n$, $\{p_1, \ldots, p_\ell\} \sim \{\pi(\{x_{1,i},\, i \in I_1\}, \ldots, \{x_{n,i},\, i \in I_n\})\}$.*

This notion of simulability leads to the following.

Definition 4 (d-Non-interference). *A (u, v)-gadget C for a function over $\mathbb{K}^n$ is* d-non-interfering *(or d-NI) if and only if any set of at most d probes of C is t-simulable, $t \leq d$.*

Definition 5 (d-Strong non-interference). *A (u, v)-gadget C for a function over $\mathbb{K}^n$ is* d-strong non-interfering *(or d-SNI) if and only if for every set P_1 of at most d_1* internal probes *(that do not depend on "output wires" or output shares $y_{i,j}$'s) and every set P_2 of d_2* external probes *(on output wires or shares) such that $d_1 + d_2 \leq d$, then $P_1 \cup P_2$ is d_1-simulable.*

It is clear that a d-SNI gadget is also d-NI. Barthe *et al.* also showed that the two notions were not equivalent, but that the composition of a d-NI and a d-SNI gadget was d-SNI [BBD+16].

3 The masking schemes of CRYPTO 2017

We recall here the main ideas of the two masking schemes of Belaïd *et al.* introduced at CRYPTO 2017 [BBP+17] and their associated matrix conditions; we refer to that paper for a full description of the gadgets and algorithms.

3.1 Pseudo-linear multiplication complexity [BBP+17, §4]

This scheme is the composition of two gadgets, only the first of which is of interest to us. In order to build a d-SNI multiplication gadget with $d+1$ input and output shares, Belaïd *et al.* first give a d-NI gadget with $d+1$ input and $2d+1$ output shares, and then compress its output into $d+1$ shares using a d-SNI gadget from Carlet *et al.* [CPRR16].

To implement d-NI multiplication over a field $\mathbb{K}$, the first gadget needs a certain matrix $\boldsymbol{\gamma} \in \mathbb{K}^{d\times d}$; in turn, this defines a related matrix $\boldsymbol{\delta} \in \mathbb{K}^{d\times d}$ as $\boldsymbol{\delta} = \mathbf{1}_{d\times d} - \boldsymbol{\gamma}$. The multiplication algorithm is then derived from the equality:

$$\begin{aligned} a \cdot b = &\left(a_0 + \sum_{i=1}^{d}(r_i + a_i)\right) \cdot \left(b_0 + \sum_{i=1}^{d}(s_i + b_i)\right) \\ &- \sum_{i=1}^{d} r_i \cdot \left(b_0 + \sum_{j=1}^{d}(\boldsymbol{\delta}_{i,j} s_j + b_j)\right) - \sum_{i=1}^{d} s_i \cdot \left(a_0 + \sum_{j=1}^{d}(\boldsymbol{\gamma}_{i,j} r_j + a_j)\right), \end{aligned}$$

where $a = \sum_{i=0}^{d} a_i$, $b = \sum_{i=0}^{d} b_i$ are the shared multiplicands, and the r_is and s_is are arbitrary (*a priori* random) values. This equality leads to defining the output shares of this first gadget as:

- $c_0 := \left(a_0 + \sum_{i=1}^{d}(r_i + a_i)\right) \cdot \left(b_0 + \sum_{i=1}^{d}(s_i + b_i)\right)$;
- $c_i := -r_i \cdot \left(b_0 + \sum_{j=1}^{d}(\boldsymbol{\delta}_{i,j} s_j + b_j)\right)$, $1 \le i \le d$;
- $c_{i+d} := -s_i \cdot \left(a_0 + \sum_{j=1}^{d}(\boldsymbol{\gamma}_{i,j} r_j + a_j)\right)$, $1 \le i \le d$.

By considering a proper scheduling of the operations needed to compute the above shares and the probes that this makes available to the adversary, Belaïd *et al.* show that a necessary and sufficient condition for their resulting scheme to be d-SNI is that $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ *both* satisfy a certain condition, stated below.

Condition 4.1 ([BBP+17]). *Let $\boldsymbol{\gamma} \in \mathbb{K}^{d\times d}$; $\ell = 2d^2 + 4d + 1$; $\boldsymbol{D}_{\boldsymbol{\gamma},j} \in \mathbb{K}^{d\times d}$ be the diagonal matrix whose non-zero entry at row i is equal to $\boldsymbol{\gamma}_{j,i}$; $\boldsymbol{T}_d \in \mathbb{K}^{d\times d}$ be the upper-triangular matrix whose non-zero entries are all one; and $\boldsymbol{T}_{\boldsymbol{\gamma},j} \in \mathbb{K}^{d\times d} = \boldsymbol{D}_{\boldsymbol{\gamma},j}\boldsymbol{T}_d$. Equivalently:*

$$\boldsymbol{I}_d = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{D}_{\boldsymbol{\gamma},j} = \begin{pmatrix} \boldsymbol{\gamma}_{j,1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\gamma}_{j,2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \boldsymbol{\gamma}_{j,d} \end{pmatrix},$$

$$\boldsymbol{T}_d = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{T}_{\boldsymbol{\gamma},j} = \begin{pmatrix} \boldsymbol{\gamma}_{j,1} & \boldsymbol{\gamma}_{j,1} & \cdots & \boldsymbol{\gamma}_{j,1} \\ 0 & \boldsymbol{\gamma}_{j,2} & \cdots & \boldsymbol{\gamma}_{j,2} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \boldsymbol{\gamma}_{j,d} \end{pmatrix}.$$

One then defines $\boldsymbol{L} \in \mathbb{K}^{(d+1)\times \ell}$ *and* $\boldsymbol{M}_{\boldsymbol{\gamma}} \in \mathbb{K}^{d\times \ell}$ *as:*

$$\boldsymbol{L} = \begin{pmatrix} 1 & \mathbf{0}_{1\times d} & \mathbf{0}_{1\times d} & \mathbf{0}_{1\times d} & \mathbf{0}_{1\times d} & \cdots & \mathbf{0}_{1\times d} & \mathbf{1}_{1\times d} & \mathbf{1}_{1\times d} & \cdots & \mathbf{1}_{1\times d} \\ \mathbf{0}_{d\times 1} & \boldsymbol{I}_d & \mathbf{0}_{d\times d} & \boldsymbol{I}_d & \boldsymbol{I}_d & \cdots & \boldsymbol{I}_d & \boldsymbol{T}_d & \boldsymbol{T}_d & \cdots & \boldsymbol{T}_d \end{pmatrix},$$

$$\boldsymbol{M}_{\boldsymbol{\gamma}} = \begin{pmatrix} \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times d} & \boldsymbol{I}_d & \boldsymbol{I}_d & \boldsymbol{D}_{\boldsymbol{\gamma},1} & \cdots & \boldsymbol{D}_{\boldsymbol{\gamma},d} & \boldsymbol{T}_d & \boldsymbol{T}_{\boldsymbol{\gamma},1} & \cdots & \boldsymbol{T}_{\boldsymbol{\gamma},d} \end{pmatrix}.$$

Finally, $\boldsymbol{\gamma}$ *is said to satisfy Condition 4.1 if for any vector* $\boldsymbol{v} \in \mathbb{K}^{\ell}$ *of Hamming weight* $\mathrm{wt}(\boldsymbol{v}) \leq d$ *such that* $\boldsymbol{L}\boldsymbol{v}$ *contains no zero coefficient (*i.e. *is of maximum Hamming weight* $d+1$*), then* $\boldsymbol{M}_{\boldsymbol{\gamma}}\boldsymbol{v} \neq \mathbf{0}_{d\times 1}$.

An equivalent, somewhat more convenient formulation of Condition 4.1 can be obtained by contraposition; $\boldsymbol{\gamma}$ satisfies Condition 4.1 if:

$$\boldsymbol{v} \in \ker(\boldsymbol{M}_{\boldsymbol{\gamma}}) \wedge \mathrm{wt}(\boldsymbol{v}) \leq d \Rightarrow \mathrm{wt}(\boldsymbol{L}\boldsymbol{v}) < d+1. \tag{1}$$

Whichever formulation is adopted, the logic behind this condition is that a violation of the implication means that there exists a linear combination of at most d probes that depends on all the input shares (as $\boldsymbol{L}\boldsymbol{v}$ is of full weight) and on no random mask (as $\boldsymbol{M}_{\boldsymbol{\gamma}}\boldsymbol{v} = \mathbf{0}_{d\times 1}$). In that respect, $\boldsymbol{L}$ and $\boldsymbol{M}$ behave as "indicator matrices" for the shares and masks on which depend individual probes.

3.2 Linear randomness complexity [BBP+17, §5]

The second scheme that we consider is defined by a single d-NI multiplication gadget over $\mathbb{K}$ that has $(d+1)$ input and output shares. An instantiation depends on a matrix $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ whose rows sum to zero, *i.e.*, such that $\sum_{i=0}^{d} \boldsymbol{\gamma}_i = \mathbf{0}_{1\times d}$.[‡] This lets us defining the output shares as:

- $c_i = a_0 b_i + \sum_{j=1}^{d} (\boldsymbol{\gamma}_{i,j} r_j + a_j b_i)$, $0 \leq i \leq d$,

where again $a = \sum_{i=0}^{d} a_i$, $b = \sum_{i=0}^{d} b_i$ are the shared multiplicands and the r_is are arbitrary values.

Belaïd *et al.* show that a necessary and sufficient condition for their resulting gadget to be d-NI is that $\boldsymbol{\gamma}$ satisfies a condition similar to Condition 4.1, stated below.

Condition 5.1 ([BBP+17]). *Let* $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ ℓ, $\boldsymbol{D}_{\boldsymbol{\gamma},j}$, $\boldsymbol{T}_d$, $\boldsymbol{T}_{\boldsymbol{\gamma},j}$ *be as in Condition 4.1 and* $\mathbb{K}(\omega_0, \ldots, \omega_d)$ *be the field of rational fractions over indeterminates* $\omega_0, \ldots, \omega_d$*; define* $\boldsymbol{L'} \in \mathbb{K}(\omega_0, \ldots, \omega_d)^{(d+1)\times \ell}$ *and* $\boldsymbol{M}'_{\boldsymbol{\gamma}} \in \mathbb{K}^{d\times \ell}$ *as:*

$$\boldsymbol{L'} = \begin{pmatrix} 1 & \mathbf{0}_{1\times d} & \mathbf{0}_{1\times d} & \mathbf{0}_{1\times d} & \mathbf{0}_{1\times d} & \cdots & \mathbf{0}_{1\times d} & \omega_0\mathbf{1}_{1\times d} & \omega_1\mathbf{1}_{1\times d} & \cdots & \omega_d\mathbf{1}_{1\times d} \\ \mathbf{0}_{d\times 1} & \boldsymbol{I}_d & \mathbf{0}_{d\times d} & \omega_0\boldsymbol{I}_d & \omega_1\boldsymbol{I}_d & \cdots & \omega_d\boldsymbol{I}_d & \omega_0\boldsymbol{T}_d & \omega_1\boldsymbol{T}_d & \cdots & \omega_d\boldsymbol{T}_d \end{pmatrix},$$

$$\boldsymbol{M}'_{\boldsymbol{\gamma}} = \begin{pmatrix} \mathbf{0}_{d\times 1} & \mathbf{0}_{d\times d} & \boldsymbol{I}_d & \boldsymbol{D}_{\boldsymbol{\gamma},0} & \boldsymbol{D}_{\boldsymbol{\gamma},1} & \cdots & \boldsymbol{D}_{\boldsymbol{\gamma},d} & \boldsymbol{T}_{\boldsymbol{\gamma},0} & \boldsymbol{T}_{\boldsymbol{\gamma},1} & \cdots & \boldsymbol{T}_{\boldsymbol{\gamma},d} \end{pmatrix}.$$

Then $\boldsymbol{\gamma}$ *is said to satisfy Condition 5.1 if for any vector* $\boldsymbol{v} \in \mathbb{K}^{\ell}$ *of Hamming weight* $\mathrm{wt}(\boldsymbol{v}) \leq d$ *such that* $\boldsymbol{L'}\boldsymbol{v}$ *contains no zero coefficient, then* $\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{v} \neq \mathbf{0}_{d\times 1}$.

[‡] Note that for convenience in the subsequent share definitions and consistency with the notation of [BBP+17], the row index of $\boldsymbol{\gamma}$ starts from zero and not one.

Note that as $\mathbb{K}$ is a subfield of $\mathbb{K}(\omega_0, \ldots, \omega_d)$ (*viz.* the field of its constants), the product $\boldsymbol{L'v}$ is well-defined. Also, again by contraposition, Condition 5.1 can be expressed as:

$$\boldsymbol{v} \in \ker(\boldsymbol{M'_\gamma}) \wedge \mathrm{wt}(\boldsymbol{v}) \leq d \Rightarrow \mathrm{wt}(\boldsymbol{L'v}) < d + 1. \tag{2}$$

4 Simplifying and unifying the conditions

In this section, we describe a few simplifications and consolidations of the correctness and safety for the two schemes described in the previous section. These simplifications are important for our analytical and algorithmic results, and the consolidations of the two schemes allow for ease in presentation.

Specifically, we develop three related conditions $\mathcal{C}$, $\mathcal{C}'$, and $\mathcal{C}''$, on the matrices $\boldsymbol{M_\gamma}$, $\boldsymbol{L}_d$, $\boldsymbol{M'_\gamma}$, and $\boldsymbol{L'}_d$ defined in Conditions 4.1 and 5.1, such that the safety of the masking schemes is guaranteed when these conditions are true. We prove that the first condition $\mathcal{C}$ and the third condition $\mathcal{C}''$ are both exactly equivalent to the requirements of Conditions 4.1 and 5.1. The second condition $\mathcal{C}'$ is always a *sufficient* condition as it implies the other two, and it is also *necessary* under a very mild condition on the cardinality of $\mathbb{K}$.

4.1 Unifying $\boldsymbol{M_\gamma}$ and $\boldsymbol{M'_\gamma}$

Recall the definitions of matrices $\boldsymbol{M_\gamma}$ from Condition 4.1 and $\boldsymbol{M'_\gamma}$ from Condition 5.1. These are both $d \times \ell$ matrices (where $\ell = 2d^2 + 4d + 1$) consisting of zeros, ones, and entries from $\boldsymbol{\gamma}$. Moreover, $\boldsymbol{M_\gamma}$ and $\boldsymbol{M'_\gamma}$ are exactly the same except for in one submatrix of d columns: this submatrix is $\boldsymbol{T}_d$ in $\boldsymbol{M_\gamma}$ and $\boldsymbol{T}_{\boldsymbol{\gamma},0}$ in $\boldsymbol{M'_\gamma}$.

We can unify these two matrices by considering, in the case of Condition 4.1, augmenting the $\boldsymbol{\gamma}$ matrix with an additional row of 1's at index 0. Then $\boldsymbol{T}_d = \boldsymbol{T}_{\boldsymbol{\gamma},0}$ and we can consider only the second form of the matrix $\boldsymbol{M'_\gamma}$.

Note that the corresponding matrices $\boldsymbol{L_\gamma}$ and $\boldsymbol{L'_\gamma}$ from Conditions 4.1 and 5.1 respectively are still not identical, but the locations of non-zero entries (i.e., the *support*) in $\boldsymbol{L_\gamma}$ and $\boldsymbol{L'_\gamma}$ are the same.

Now for both schemes, there is a single matrix $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ which determines their *correctness* (do the output shares always correspond to the multiplication of the input value) and *safety* (is it possible for an attacker to learn any secret with at most d probes).

To succinctly state the unified condition, we first define a simple predicate $\mathcal{Z}$ for when a matrix $\boldsymbol{X} \in \mathbb{K}^{m\times n}$ (or column vector $\boldsymbol{x} \in \mathbb{K}^m$) has at least one row of zeros:

$$\mathcal{Z}(\boldsymbol{X}) := \exists\, i \in \{1, \ldots, m\} \text{ s.t. } \forall\, j \in \{1, \ldots, n\}, \boldsymbol{X}_{i,j} = 0.$$

Based on the above discussion, we define the following crucial predicate for the safety definition for two arbitrary matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ with the same number

of columns:

$$\mathcal{C}(\boldsymbol{A}, \boldsymbol{B}) := \forall\, \boldsymbol{v} \in \ker(\boldsymbol{A}) \text{ s.t. } \mathrm{wt}(\boldsymbol{v}) \leq \mathrm{rowdim}(\boldsymbol{A}),\ \text{then } \mathcal{Z}(\boldsymbol{B}\boldsymbol{v}). \tag{3}$$

Typically we will have $\boldsymbol{A} = \boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{B}$ is either $\boldsymbol{L}$ or $\boldsymbol{L}'$.

Now we can restate the correctness and safety conditions for the two schemes. The following propositions follow directly from the definitions and discussions so far.

Proposition 6. *For $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$, the scheme of Section 3.1 is correct and safe if and only if the following conditions are met, where $\boldsymbol{\delta} = \begin{pmatrix} \mathbf{2}_{1\times d}{}^{\S} \\ \mathbf{1}_{d\times d} \end{pmatrix} - \boldsymbol{\gamma}$:*

(1) $\boldsymbol{\gamma}_{0,j} = 1$ *for all* $j \in \{1, \ldots, d\}$
(2) $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{L})$
(3) $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\delta}}, \boldsymbol{L})$

Proposition 7. *For $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$, the scheme of Section 3.2 is correct and safe if and only if the following conditions are met:*

(1) $\sum_{i=0}^{d} \boldsymbol{\gamma}_i = \mathbf{0}_{1\times d}$

(2) $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{L}')$

4.2 Equivalent condition with kernel bases

Next we develop a condition similar to the definition of $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ as defined in (3) above, but in terms of kernel bases rather than individual vectors. This modified condition is equivalent under a mild requirement on the size of the field $\mathbb{K}$.

The general idea is that rather than considering all matrix-vector products $\boldsymbol{B}\boldsymbol{v}$, where $\boldsymbol{v}$ is a d-sparse vector in the right kernel of $\boldsymbol{A}$, we consider instead the kernel basis for a size-d subset of $\boldsymbol{A}$'s columns, and multiply the corresponding columns in $\boldsymbol{B}$ times this basis. Specifying this condition requires some additional notation which will also be useful later on.

Let $\mathrm{kerb}(\boldsymbol{X})$ denote a basis of the right kernel of $\boldsymbol{X}$. That is, any vector $\boldsymbol{v} \in \ker(\boldsymbol{X})$ is a linear combination of the columns of $\mathrm{kerb}(\boldsymbol{X})$.

Let $[c_1, \ldots, c_k]$ be a list of k distinct column indices, where each $1 \leq c_i \leq \ell$. Selecting only these columns from any matrix with ℓ columns is a linear operator corresponding to a *selection matrix* $\boldsymbol{P} \in \{0,1\}^{\ell\times k}$, where $\boldsymbol{P}_{i,j} = 1$ iff $c_j = i$. Define S^{ℓ}_{m} as the set of all $\ell \times m$ selection matrices. That is, S^{ℓ}_{m} consists of all $\{0,1\}$-matrices with ℓ rows and at most m columns, where there is a single 1 in each column and no two 1s in the same row.

Note that the product of a selection matrix and its transpose is an identity matrix with some rows and columns set to zero. For any matrix (or vector) $\boldsymbol{X} \in \mathbb{K}^{m\times n}$ with at most k non-zero rows, there is a selection matrix $\boldsymbol{P} \in S^{k}_{m}$ such that $\boldsymbol{P}\boldsymbol{P}^{T}\boldsymbol{X} = \boldsymbol{X}$.

[§] In fields of characteristic 2, the matrix $\mathbf{2}_{1\times d}$ is actually $\mathbf{0}_{1\times d}$.

The equivalent condition to (3) that we consider now is formed by multiplying some subset of $\boldsymbol{B}$'s columns times a kernel basis of the same subset of $\boldsymbol{A}$'s columns:

$$\mathcal{C}'(\boldsymbol{A}, \boldsymbol{B}) \;:=\; \forall\, \boldsymbol{P} \in S^{\ell}_{\mathrm{rowdim}(\boldsymbol{A})},\; \mathcal{Z}(\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP})). \tag{4}$$

One direction of the equivalence is straightforward, and the other depends on the Schwartz-Zippel lemma and therefore on the size of the field. Even so, the field size requirement here is very mild; indeed the field is sufficiently large in all cases where we are aware of any valid constructions of the schemes.

Theorem 8. *For any $\boldsymbol{A} \in \mathbb{K}^{n \times \ell}$ and $\boldsymbol{B} \in \mathbb{K}^{m \times \ell}$, we have $\mathcal{C}'(\boldsymbol{A}, \boldsymbol{B}) \Rightarrow \mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$. If $\mathbb{K}$ has at least $m + 1$ distinct elements, then $\mathcal{C}'(\boldsymbol{A}, \boldsymbol{B}) \Leftarrow \mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ also.*

Proof. We begin with the "$\Rightarrow$" direction.

Let $\boldsymbol{v}$ be a vector satisfying the conditions of $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$; that is, $\boldsymbol{v} \in \ker \boldsymbol{A}$ and $\mathrm{wt}(\boldsymbol{v}) \leq \mathrm{rowdim}(\boldsymbol{A})$. The latter fact means that there exists $\boldsymbol{P} \in S^{\ell}_{\mathrm{rowdim}(\boldsymbol{A})}$ such that $\boldsymbol{PP}^T\boldsymbol{v} = \boldsymbol{v}$.

Because $\boldsymbol{Av} = \boldsymbol{0}$, we then have $(\boldsymbol{AP})(\boldsymbol{P}^T\boldsymbol{v}) = \boldsymbol{0}$, which means that the vector $\boldsymbol{P}^T\boldsymbol{v}$ is a linear combination of the columns of $\mathrm{kerb}(\boldsymbol{AP})$.

The condition $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ concerns the matrix-vector product $\boldsymbol{Bv}$, which equals $\boldsymbol{BPP}^T\boldsymbol{v}$. From above, we know that this is a linear combination of the columns in the matrix $\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP})$. By the assumption that $\mathcal{C}'(\boldsymbol{A}, \boldsymbol{B})$, this matrix contains a zero row, and therefore any linear combination of its columns also contains a zero row; hence $\mathcal{Z}(\boldsymbol{Bv})$.

For the "$\Leftarrow$" direction, we prove using the contrapositive. Assume there exists some selection of columns $\boldsymbol{P} \in S^{\ell}_{n}$ such that $\neg\, \mathcal{Z}(\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP}))$. We need to show that $\neg\, \mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$.

Suppose the column dimension of $\mathrm{kerb}(\boldsymbol{AP})$ (*i.e.*, the nullity of $\boldsymbol{AP}$) is k, and let $\boldsymbol{x}$ be a column vector of k indeterminates $x_1, \ldots, x_k$. Now consider the matrix-vector product $\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP}) \cdot \boldsymbol{x}$. This is a column vector of dimension m consisting of degree-1 polynomials in the k indeterminates. Furthermore, none of these polynomials is zero because of the assumption $\neg\, \mathcal{Z}(\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP}))$.

The product of the m polynomials in $\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP}) \cdot \boldsymbol{x}$ is a single non-zero polynomial in k variables with total degree m. By the Schwartz-Zippel-DeMillo-Lipton lemma [Sch80, Cor. 1], and because $\#\mathbb{K} > m$, there must exist some assignment of the k variables to values in $\mathbb{K}$ such that this product polynomial is non-zero. That is, there exists some column vector $\boldsymbol{w} \in \mathbb{K}^k$ such that $\mathrm{wt}(\boldsymbol{BP} \cdot \mathrm{kerb}(\boldsymbol{AP}) \cdot \boldsymbol{w}) = m$.

Because $\mathrm{kerb}(\boldsymbol{AP}) \cdot \boldsymbol{w} \in \mathbb{K}^n$, there is an n-sparse vector $\boldsymbol{v} \in \mathbb{K}^{\ell}$ such that $\boldsymbol{P}^T\boldsymbol{v} = \mathrm{kerb}(\boldsymbol{AP}) \cdot \boldsymbol{w}$. This vector $\boldsymbol{v}$ shows that $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ is false. Namely, $\boldsymbol{v} \in \ker(\boldsymbol{A})$ because $\boldsymbol{Av} = (\boldsymbol{AP})(\boldsymbol{P}^T\boldsymbol{v}) = \boldsymbol{0}$; it has low weight $\mathrm{wt}(\boldsymbol{v}) \leq n$; and $\boldsymbol{Bv} = (\boldsymbol{BP})(\boldsymbol{P}^T\boldsymbol{v})$ is of full weight m from the previous paragraph. □

4.3 Eliminating rows and columns

The third simplification to the correctness and safety conditions of the two masking schemes that we develop is an equivalent condition to $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ that depends

on less than half of the columns in the original matrices. The intuition is that most of the columns of these matrices have weight 1, and thus those probes in the masking scheme do not gain the attacker any real advantage. So we can focus on only the parts of $\boldsymbol{A}$ and $\boldsymbol{B}$ whose columns have weight greater than 1. We first develop some new terminology to talk about these submatrices, then prove a lemma which shows how to eliminate columns from $\boldsymbol{\gamma}$ corresponding to the weight-one probes, and finally state and prove the equivalent condition $\mathcal{C}''$.

So far the schemes are both defined by a matrix $\boldsymbol{\gamma}$ with $d+1$ rows and d columns. In fact, the definitions of matrices $\boldsymbol{M_\gamma}$, $\boldsymbol{M'_\gamma}$, $\boldsymbol{L}$, and $\boldsymbol{L'}$ from Conditions 4.1 and 5.1 generalize to any rectangular matrix $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times n}$. If $\boldsymbol{\gamma}$ has $d+1$ rows and n columns, then $\boldsymbol{M_\gamma}$ and $\boldsymbol{M'_\gamma}$ both have n rows, while $\boldsymbol{L}_n$ and $\boldsymbol{L}'_n$ have $n+1$ rows, and all four matrices have $\ell_n = 2dn + 4n + 1$ columns.

We focus on the bottom-right $n \times (dn+n)$ submatrix of each $\boldsymbol{M'_\gamma}$, $\boldsymbol{L}_n$ and $\boldsymbol{L}'_n$, which we call the "triangular part" of each. Formally, we define a linear operator $\boldsymbol{\Delta}$ such that, for any matrix $\boldsymbol{A}$ with n or $n+1$ rows and $2dn+4n+1$ columns, $\boldsymbol{\Delta}(\boldsymbol{A})$ consists of the bottom-right $n \times (dn+n)$ submatrix of $\boldsymbol{A}$.

In summary, we have:

$$\boldsymbol{L}_n = \begin{pmatrix} 1 & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \cdots & \mathbf{0}_{1\times n} & \underbrace{\mathbf{1}_{1\times n} & \mathbf{1}_{1\times n} & \cdots & \mathbf{1}_{1\times n}}_{\boldsymbol{\Delta}(\boldsymbol{L}_n)} \\ \mathbf{0}_{n\times 1} & \boldsymbol{I}_n & \mathbf{0}_{n\times n} & \boldsymbol{I}_n & \boldsymbol{I}_n & \cdots & \boldsymbol{I}_n & \boldsymbol{T}_n & \boldsymbol{T}_n & \cdots & \boldsymbol{T}_n \end{pmatrix},$$

$$\boldsymbol{L}'_n = \begin{pmatrix} 1 & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \mathbf{0}_{1\times n} & \cdots & \mathbf{0}_{1\times n} & \underbrace{\omega_0\mathbf{1}_{1\times n} & \omega_1\mathbf{1}_{1\times n} & \cdots & \omega_d\mathbf{1}_{1\times n}}_{\boldsymbol{\Delta}(\boldsymbol{L}'_n)} \\ \mathbf{0}_{n\times 1} & \boldsymbol{I}_n & \mathbf{0}_{n\times n} & \omega_0\boldsymbol{I}_n & \omega_1\boldsymbol{I}_n & \cdots & \omega_d\boldsymbol{I}_n & \omega_0\boldsymbol{T}_n & \omega_1\boldsymbol{T}_n & \cdots & \omega_d\boldsymbol{T}_n \end{pmatrix},$$

$$\boldsymbol{M'_\gamma} = \begin{pmatrix} \mathbf{0}_{n\times 1} & \mathbf{0}_{n\times n} & \boldsymbol{I}_n & \boldsymbol{D}_{\boldsymbol{\gamma},0} & \boldsymbol{D}_{\boldsymbol{\gamma},1} & \cdots & \boldsymbol{D}_{\boldsymbol{\gamma},d} & \underbrace{\boldsymbol{T}_{\boldsymbol{\gamma},0} & \boldsymbol{T}_{\boldsymbol{\gamma},1} & \cdots & \boldsymbol{T}_{\boldsymbol{\gamma},d}}_{\boldsymbol{\Delta}(\boldsymbol{M'_\gamma})} \end{pmatrix}.$$

Notice that the matrices $\boldsymbol{L}_n$ and $\boldsymbol{L}'_n$ have some different entries but the same support; for convenience we denote by $\boldsymbol{N}_n$ any matrix with this same dimension and support.

Inspecting the definition of $\boldsymbol{M'_\gamma}$, we see that rows of this matrix correspond to columns of $\boldsymbol{\gamma}$, and removing one column of $\boldsymbol{\gamma}$ corresponds to removing a single row and $2d+4$ columns from each of $\boldsymbol{M'_\gamma}$ and $\boldsymbol{N}$.

Notice also that the columns of $\boldsymbol{M'_\gamma}$ and of $\boldsymbol{L}_n$ which are not in the triangular parts all have weight at most one. This means, as we show in the following technical lemma, that the effect of any such column choice (as a probe) can be eliminated by removing one row each from $\boldsymbol{M'_\gamma}$ and $\boldsymbol{L}_n$. In terms of masking schemes, this means that a single probe corresponding to these non-triangular parts allows the adversary to cancel at most one random value and to learn at most one share. Because the number of shares is $d+1$ in a scheme allowing d probes, this results in no advantage for the adversary.

Lemma 9. *Let $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times n}$, $\boldsymbol{M'_\gamma}$ and $\boldsymbol{N}_n$ be as above. Suppose $\boldsymbol{u} \in \mathbb{K}^{\ell_n}$ is a vector with $\mathrm{wt}(\boldsymbol{u}) = 1$ whose single non-zero entry is between index 2 and*

$dn + 3n + 1$ inclusive, and $\boldsymbol{v} \in \mathbb{K}^{\ell_n}$ is any other vector. Then there exists a selection matrix $\boldsymbol{P} \in S^n_{n-1}$ and another vector $\boldsymbol{w} \in \mathbb{K}^{\ell_{n-1}}$ with $\mathrm{wt}(\boldsymbol{w}) \leq \mathrm{wt}(\boldsymbol{v})$ such that

$$\mathrm{wt}(\boldsymbol{M}'_{\boldsymbol{\gamma P}}\boldsymbol{w}) \leq \mathrm{wt}(\boldsymbol{M}'_{\boldsymbol{\gamma}}(\boldsymbol{u}+\boldsymbol{v})) \quad \textit{and} \quad \mathrm{wt}(\boldsymbol{N}_{n-1}\boldsymbol{w}) \geq \mathrm{wt}(\boldsymbol{N}_n(\boldsymbol{u}+\boldsymbol{v})) - 1.$$

Proof. Write i for the index of the non-zero entry in $\boldsymbol{u}$. We can see that the ith column of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$ both have weight at most one. Indeed, for each $i \in \{2, \ldots, dn+3n+1\}$, there is a corresponding index $j \in \{1, \ldots, n\}$ such that the ith columns of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$ are zero everywhere except possibly in row j (provided that we continue to index the rows of $\boldsymbol{N}_n$ starting at 0).

Removing the jth row from $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$ results in two new matrices $\boldsymbol{A}, \boldsymbol{B}$ (respectively) whose ith columns are both zero, and hence $\boldsymbol{Au} = \boldsymbol{0}$ and $\boldsymbol{Bu} = \boldsymbol{0}$. This means that

$$\begin{aligned}\mathrm{wt}(\boldsymbol{Av}) &= \mathrm{wt}(\boldsymbol{A}(\boldsymbol{u}+\boldsymbol{v})) \leq \mathrm{wt}(\boldsymbol{M}'_{\boldsymbol{\gamma}}(\boldsymbol{u}+\boldsymbol{v}))\\ \mathrm{wt}(\boldsymbol{Bv}) &= \mathrm{wt}(\boldsymbol{B}(\boldsymbol{u}+\boldsymbol{v})) \geq \mathrm{wt}(\boldsymbol{N}_n(\boldsymbol{u}+\boldsymbol{v})) - 1.\end{aligned}$$

Write $\boldsymbol{P} \in S^n_{n-1}$ as the matrix which selects all n columns of $\boldsymbol{\gamma}$ except for the jth column. Now $\boldsymbol{A}$ and $\boldsymbol{B}$ are the same as $\boldsymbol{M}'_{\boldsymbol{\gamma P}}$ and $\boldsymbol{N}_{n-1}$ respectively, except that they each have $2d+4$ extra columns. The remaining task is to modify $\boldsymbol{v}$ so that it is zero at all the indices corresponding to these extra columns, without changing $\mathrm{wt}(\boldsymbol{Av})$ or $\mathrm{wt}(\boldsymbol{Bv})$.

We can see that $d+3$ of these extra columns come from the first $dn+3n+1$ columns of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$ and, since the jth row has been removed, they are in fact now zero columns. So letting $\boldsymbol{v}'$ be the same as $\boldsymbol{v}$ with any such entries set to zero, we do not change the products $\boldsymbol{Av}'$ or $\boldsymbol{Bv}'$ at all.

The $d+1$ remaining extra columns come from the triangular parts $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ and $\boldsymbol{\Delta}(\boldsymbol{N}_n)$. There are now two cases to consider. First, if $j = 1$, *i.e.*, we have removed the second row of $\boldsymbol{N}_n$ and the first row of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$. Then these extra columns from the triangular part of $\boldsymbol{A}$ are all zero columns, and from $\boldsymbol{B}$ they have the form $(a\ 0\ \cdots\ 0)^T$ for some non-zero entry a in the first row of $\boldsymbol{N}_n$. Upon inspection, we see that these columns are exactly a times the very first columns of $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively. Therefore we can modify the vector $\boldsymbol{v}'$ to a new vector $\boldsymbol{v}''$, where any non-zero entries in such positions are divided by a and added to the first entry, then set to zero. This does not change the value of $\boldsymbol{Av}''$ or $\boldsymbol{Bv}''$.

The second case is that $j \geq 2$, *i.e.*, we have removed a later row. Then the extra columns in $\boldsymbol{A}$ and $\boldsymbol{B}$ are exactly identical to the columns immediately to their left in the respective matrices. So we can form $\boldsymbol{v}''$ in this case by adding any non-zero entry of $\boldsymbol{v}'$ in such positions to the adjacent position and then setting it to zero, without changing $\boldsymbol{Av}''$ or $\boldsymbol{Bv}''$.

After this, we have a vector $\boldsymbol{v}''$ with $\mathrm{wt}(\boldsymbol{v}'') \leq \mathrm{wt}(\boldsymbol{v})$, and with zeros in all of the "extra column" indices of $\boldsymbol{A}$ and $\boldsymbol{B}$, such that $\mathrm{wt}(\boldsymbol{Av}'') \leq \mathrm{wt}(\boldsymbol{M}'_{\boldsymbol{\gamma}}(\boldsymbol{u}+\boldsymbol{v}))$ and $\mathrm{wt}(\boldsymbol{Bv}'') \geq \mathrm{wt}(\boldsymbol{N}_n(\boldsymbol{u}+\boldsymbol{v})) - 1$. Finally, setting $\boldsymbol{w}$ to be the sub-vector of $\boldsymbol{v}''$ with these extra column entries removed completes the proof. □

Repeated application of the previous lemma allows us to completely eliminate all of the columns in $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$ other than the triangular parts, at the cost

of having to consider all possible column-subsets of $\boldsymbol{\gamma}$ itself. This leads to the following condition:

$$\mathcal{C}''(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n) := \ \forall\, k \in \{1, \ldots, n\}, \forall\, \boldsymbol{P} \in S_k^n,\ \mathcal{C}(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}\boldsymbol{P}}), \boldsymbol{\Delta}(\boldsymbol{N}_k)). \tag{5}$$

In other words, we restrict our attention to only square submatrices of the triangular parts of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$. As it turns out, this condition is exactly equivalent to the original one.

Theorem 10. *For any field $\mathbb{K}$, matrix $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times n}$ where $n \geq 1$, and matrix $\boldsymbol{N}_n \in \{\boldsymbol{L}_n, \boldsymbol{L}'_n\}$, we have $\mathcal{C}''(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n) \Leftrightarrow \mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$.*

Proof. We prove the equivalent double negation $\neg\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n) \Leftrightarrow \neg\mathcal{C}''(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$.

First we prove the "$\Rightarrow$" direction by induction on n. Assuming that $\neg\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$ means there exists a vector $\boldsymbol{v} \in \mathbb{K}^{\ell_n}$ such that $\mathrm{wt}(\boldsymbol{v}) \leq n$, $\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{v} = \boldsymbol{0}$, and $\boldsymbol{N}_n\boldsymbol{v}$ has full weight $n+1$.

For the base case, let $n = 1$. Because $\mathrm{wt}(\boldsymbol{v}) = 1$ and $\mathrm{wt}(\boldsymbol{N}_n\boldsymbol{v}) = 2$, the lone non-zero entry of $\boldsymbol{v}$ must correspond to a weight-2 column in $\boldsymbol{N}_n$, and the only such columns are in the triangular part. So considering the vector formed from the last $d+1$ entries of $\boldsymbol{v}$ shows that $\neg\mathcal{C}(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_n))$, which is equivalent to $\neg\mathcal{C}''(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$ when $n = 1$.

Now for the induction case, let $n \geq 2$ and assume the $\Rightarrow$ direction is true for all size-$(n-1)$ subsets of columns of $\boldsymbol{\gamma}$.

Again we start with a vector $\boldsymbol{v}$ which is a counterexample to $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$. If $\boldsymbol{v}$ has any non-zero entry in indices 2 through $dn+3n+1$, then we can isolate that entry in its own vector $\boldsymbol{u}$ and write $\boldsymbol{v} = \boldsymbol{u}+\boldsymbol{v}^*$, where $\mathrm{wt}(\boldsymbol{v}^*) = \mathrm{wt}(\boldsymbol{v})-1 \leq n-1$. Now apply Lemma 9 to obtain a vector $\boldsymbol{w} \in \mathbb{K}^{\ell_{n-1}}$ and a selection matrix $\boldsymbol{P} \in S_{n-1}^n$ such that $\mathrm{wt}(\boldsymbol{w}) \leq n-1$, $\boldsymbol{M}'_{\boldsymbol{\gamma}\boldsymbol{P}}\boldsymbol{w} = \boldsymbol{0}$, and $\mathrm{wt}(\boldsymbol{N}_{n-1}\boldsymbol{w}) = n-1$. Therefore $\neg\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}\boldsymbol{P}}, \boldsymbol{N}_{n-1})$, so we can apply the induction hypothesis to complete this sub-case.

Otherwise, the non-zero entries of $\boldsymbol{v}$ are in the very first index, or in the last $(d+1)n$ indices which correspond to the triangular parts. But the first columns of $\boldsymbol{N}_n$ and $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ are all zeros except for the first row in $\boldsymbol{N}_n$, which is eliminated in the triangular part $\boldsymbol{\Delta}(\boldsymbol{N}_n)$. Therefore, if this entry of $\boldsymbol{v}$ is non-zero, we can change it to zero without affecting $\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{v}$, which must equal $\boldsymbol{0}$, or the last n rows of $\boldsymbol{N}_n\boldsymbol{v}$, which must be all non-zero. Hence the vector consisting of the last $(d+1)n$ entries of $\boldsymbol{v}$ is a counterexample to $\mathcal{C}(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_n))$. This completes the $\Rightarrow$ direction of the proof.

For the $\Leftarrow$ direction, assume that $\neg\mathcal{C}''(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$. This means there is some $k \in \{1, \ldots, n\}$, some selection of columns from $\boldsymbol{\gamma}$ defined by $\boldsymbol{P} \in S_k^n$, and some $\boldsymbol{v} \in \mathbb{K}^{\ell_k}$ such that $\mathrm{wt}(\boldsymbol{v}) \leq k$, $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}\boldsymbol{P}})\boldsymbol{v} = \boldsymbol{0}$, and $\boldsymbol{\Delta}(\boldsymbol{N}_k)\boldsymbol{v}$ has full weight k.

Because the triangular part is a subset of the whole, we can prepend $\boldsymbol{v}$ with $dk + 3k + 1$ zeros to obtain a vector $\boldsymbol{v}'$ such that $\boldsymbol{M}'_{\boldsymbol{\gamma}\boldsymbol{P}}\boldsymbol{v}' = \boldsymbol{0}$ and $\boldsymbol{N}_k\boldsymbol{v}'$ is non-zero everywhere except possibly in the first row. Observe that the row of $\boldsymbol{N}_k$ immediately above the triangular part is exactly identical to the top row of $\boldsymbol{\Delta}(\boldsymbol{N}_k)$, so in fact $\boldsymbol{N}_k\boldsymbol{v}'$ has full weight $k+1$.

This shows that there exists at least one $k \geq 1$ such that there exists a selection $\boldsymbol{P} \in S_k^n$ and a vector $\boldsymbol{v}'$ which is a counterexample to $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma P}}, \boldsymbol{N}_k)$. Assume now that k is the *largest* such integer.

If $k = n$, then $\boldsymbol{M}'_{\boldsymbol{\gamma P}} = \boldsymbol{M}'_{\boldsymbol{\gamma}}$, and $\boldsymbol{v}'$ is a counterexample to $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_n)$ already.

Otherwise, if $k < n$, we show that we can construct a larger selection matrix $\boldsymbol{Q}$ and corresponding vector $\boldsymbol{w}$ satisfying the conditions above, which is a contradiction to the assumption that k is the largest such value.

Construct another selection matrix $\boldsymbol{Q} \in S_{k+1}^n$ consisting of the columns selected by $\boldsymbol{P}$ plus some additional column i; for convenience write $\boldsymbol{\zeta} = \boldsymbol{\gamma Q}$. Note that $\boldsymbol{M}'_{\boldsymbol{\gamma P}}$ and $\boldsymbol{N}_k$ are submatrices of $\boldsymbol{M}'_{\boldsymbol{\zeta}}$ and $\boldsymbol{N}_{k+1}$ respectively, the latter both having exactly one more row and some number of extra columns. Therefore by extending $\boldsymbol{v}'$ to a larger vector $\boldsymbol{v}''$ by inserting zeros in the locations of these extra columns, we have that $\boldsymbol{M}'_{\boldsymbol{\zeta}}\boldsymbol{v}''$ is zero everywhere except possibly at index i, and $\boldsymbol{N}_{k+1}\boldsymbol{v}''$ is non-zero everywhere except at index i. Let a be the ith entry of $\boldsymbol{M}'_{\boldsymbol{\zeta}}\boldsymbol{v}''$ and b be the ith entry of $\boldsymbol{N}_{k+1}\boldsymbol{v}''$.

Finally, we show how to add one more entry to $\boldsymbol{v}''$ to "fix" the exceptions at index i in the previous sentence, making $a = 0$ and $b \neq 0$. There are four cases to consider:

1. If $a = 0$ and $b \neq 0$, then we are done.
2. If $a = 0$ and $b = 0$, then set the $(i+1)$th entry of $\boldsymbol{v}$ to 1; this corresponds to a column of zeros in $\boldsymbol{M}'_{\boldsymbol{\zeta}}$ and a column of the identity matrix in $\boldsymbol{N}_{k+1}$. So adding that column keeps $a = 0$ but sets b to 1.
3. If $a \neq 0$ and $b \neq 0$, then set the $(k+i+1)$th entry of $\boldsymbol{v}$ to $-a$. This entry corresponds to a column of the identity matrix in $\boldsymbol{M}'_{\boldsymbol{\zeta}}$ and a column of zeros in $\boldsymbol{N}_{k+1}$, so adding it keeps $b \neq 0$ but cancels the value of a.
4. If $a \neq 0$ and $b = 0$, then set the $(2k+i+2)$th entry of $\boldsymbol{v}$ to $-a/\boldsymbol{\zeta}_{0,i}$. This entry corresponds to a column of $\boldsymbol{D}_{\boldsymbol{\zeta},0}$ in $\boldsymbol{M}'_{\boldsymbol{\zeta}}$, and a column of either $\boldsymbol{I}_{k+1}$ or $\omega_0\boldsymbol{I}_{k+1}$ within $\boldsymbol{N}_{k+1}$, and therefore the change to $\boldsymbol{v}$ cancels out a and sets b to some non-zero value.

This newly constructed vector has weight at most $\mathrm{wt}(\boldsymbol{v}'') + 1 \leq k + 1$, and is therefore a counterexample to $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\zeta}}, \boldsymbol{N}_{k+1})$. This is a contradiction to the assumption that k was maximal, which completes the $\Leftarrow$ direction and the entire proof. □

5 A matrix precondition

We use the results of the previous two sections to develop a useful precondition for generating $\boldsymbol{\gamma}$ matrices which satisfy the safety and correctness conditions of the two schemes. This precondition guarantees the correctness conditions, and (as we will see in later sections) seems to increase the probability that a matrix satisfies the safety condition. We then show how to explicitly generate matrices which satisfy these preconditions.

5.1 Definitions

As in the previous section, let $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ be a matrix whose entries determine the correctness and safety of one of the two masking schemes according to Proposition 6 or Proposition 7. (Either $\boldsymbol{\gamma}$ must have a row equal to $\mathbf{1}$, or they must sum to $\mathbf{0}$.)

Then Theorems 8 and 10 tell us that a sufficient condition for safety is that for every square submatrix of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$, all vectors in its right kernel have at least one joint zero entry when multiplied with the corresponding submatrix of $\boldsymbol{\Delta}(\boldsymbol{N}_d)$. The general idea of the preconditions developed in this section is to *minimize the rank of this right kernel*, effectively limiting the number of possible "unsafe" vectors. In particular, when a square submatrix of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ is non-singular, then its nullity is zero and the scheme is safe with respect to that subset of rows and columns.

This suggests a strategy to increase the likelihood of a matrix leading to a safe scheme: one may try to choose $\boldsymbol{\gamma}$ in a way that ensures that $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma P}})\boldsymbol{Q}$ has a trivial kernel for as many selection matrices $\boldsymbol{P} \in S_k^d$ and $\boldsymbol{Q} \in S_k^{\ell_k}$ as possible. That is, square submatrices of the triangular part of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ should be non-singular as often as possible.

A good such choice for $\boldsymbol{\gamma}$ is to take it to be such that all its square submatrices are MDS. To justify this claim, recall from Section 2 that any square submatrix of an MDS matrix is invertible, *i.e.*, has a trivial kernel. Further, from the definition of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$, its columns consist of (partial) rows of $\boldsymbol{\gamma}$; therefore many of its submatrices are in fact (transposed) submatrices of $\boldsymbol{\gamma}$ itself.

Example 11. Consider for the case $d = 3$, the submatrix of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ given by:

$$\boldsymbol{X} = \begin{pmatrix} \gamma_{0,1} & \gamma_{1,1} & \gamma_{2,1} \\ 0 & \gamma_{1,2} & \gamma_{2,2} \\ 0 & \gamma_{1,3} & \gamma_{2,3} \end{pmatrix}.$$

(Note that in the case of Condition 4.1, $\gamma_{0,1}$ must equal 1.) If all square submatrices of $\boldsymbol{\gamma}$ are MDS, the bottom-right 2×2 submatrix of $\boldsymbol{X}$ is necessarily non-singular, and $\gamma_{0,1} \neq 0$, so therefore this entire submatrix is non-singular. This would not be the case for an arbitrary matrix $\boldsymbol{\gamma}$, even if say, one takes it to be full-rank.

We now state our two *preconditions* on the matrices used to instantiate either masking scheme. As will be clear in the remainder of this paper, these preconditions are by no means sufficient, nor necessary. Yet we will also see, both formally (in Section 6) and experimentally (in Section 8) how they may be useful.

Precondition 4.1. *A matrix* $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ satisfies Precondition 4.1 *for Condition 4.1 if it can be written as* $\boldsymbol{\gamma} = \begin{pmatrix} \mathbf{1}_{1\times d} \\ \boldsymbol{A} \end{pmatrix}$, *and both matrices* $\boldsymbol{A}$ *and* $\mathbf{1}_{d\times d} - \boldsymbol{A}$ *are row XMDS.*

Any such matrix $\boldsymbol{\gamma}$ clearly satisfies the correctness condition, which is item (1) in Proposition 6. The XMDS property also ensures that all square submatrices of $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ are non-singular, which (we expect) will make the safety conditions (2) and (3) from Proposition 6 more likely satisfied.

Precondition 5.1. *A matrix $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$* satisfies Precondition 5.1 *for Condition 5.1 if $\sum_{i=0}^{d} \boldsymbol{\gamma}_i = \mathbf{0}_{1\times d}$ and all of its square submatrices are MDS.*

Again, this precondition guarantees the correctness of the scheme, corresponding to item (1) of Proposition 7, and the non-singular submatrices make it (we expect) more likely that the safety condition, item (2), is also true.

5.2 Explicit constructions

It is relatively easy to check if a given matrix satisfies either of the above preconditions. Here we do even better, providing a direct construction for families of matrices that satisfy each of them.

Theorem 12 (Satisfying Precondition 4.1). *Let $\{x_1, \ldots, x_d, y_1, \ldots, y_d\} \in \mathbb{K}\backslash\{0\}$ be $2d$ distinct non-zero elements of $\mathbb{K}$, and define matrix $\boldsymbol{A} \in \mathbb{K}^{d\times d}$ by $\boldsymbol{A}_{i,j} = x_i/(x_i - y_j)$. Then the corresponding $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ satisfies Precondition 4.1.*

Proof. Define the row-extended Cauchy matrix $\boldsymbol{B}$ as $\boldsymbol{B}_{0,j} = 1$, $1 \le j \le d$; $\boldsymbol{B}_{i,j} = (x_i - y_j)^{-1}$, $1 \le i, j \le d$. The generalized extended matrix obtained from $\boldsymbol{B}$ by the row scaling $\boldsymbol{c} = \begin{pmatrix} 1 & x_1 & \cdots & x_d \end{pmatrix}$ is equal to $\boldsymbol{\gamma}$, and all its square submatrices are invertible by construction, hence $\boldsymbol{A}$ is row XMDS.

The matrix $\boldsymbol{C} = \mathbf{1}_{d\times d} - \boldsymbol{A}$ is given by $\left((x_i - y_j - x_i)\cdot(x_i - y_j)^{-1}\right) = \left(-y_j \cdot (x_i - y_j)^{-1}\right)$. It is a generalized Cauchy matrix with column scaling given by $\begin{pmatrix} -y_1 & \ldots & -y_d \end{pmatrix}^T$, and is then MDS. Because $0 \notin \{x_1, \ldots, x_d, y_1, \ldots, y_d\}$, one may extend $\boldsymbol{C}$ by one row on top using $x_0 = 0$, resulting in $\boldsymbol{C'}$ s.t. $\boldsymbol{C'}_{0,j} = -y_j \cdot (0 - y_j)^{-1} = 1$, $1 \le j \le d$; $\boldsymbol{C'}_{i,j} = \boldsymbol{C}_{i,j}$, $1 \le i, j \le d$. In other words,

$$\boldsymbol{C'} = \begin{pmatrix} \mathbf{1}_{1\times d} \\ \boldsymbol{C} \end{pmatrix}$$

is a generalized Cauchy matrix, whose square submatrices are all invertible by construction, hence $\boldsymbol{C} = \mathbf{1}_{d\times d} - \boldsymbol{A}$ is row XMDS. □

Theorem 13 (Satisfying Precondition 5.1). *Let $\{x_1, \ldots, x_d, x_{d+1}, y_1, \ldots, y_d\} \in \mathbb{K}$ be $2d+1$ distinct elements of $\mathbb{K}$; let $\boldsymbol{A} = \left((x_i - y_j)^{-1}\right)$; and let $\boldsymbol{c} = \begin{pmatrix} c_1 & \cdots & c_{d+1} \end{pmatrix}$ be a non-zero vector in the left kernel of $\boldsymbol{A}$. Then $\boldsymbol{\gamma} = \left(c_i \cdot (x_i - y_j)^{-1}\right)$ satisfies Precondition 5.1.*

Proof. By construction, the $d+1 \times d$ Cauchy matrix $\boldsymbol{A}$ has a left kernel of dimension one. Furthermore, any vector of this kernel that is not the null vector is of full Hamming weight, as being otherwise would imply the existence of $k \le d$ linearly-dependent rows of $\boldsymbol{A}$. The row scaling coefficients $\begin{pmatrix} c_1 & \cdots & c_{d+1} \end{pmatrix}$ are thus all non-zero, and the generalized Cauchy matrix $\boldsymbol{A'}$ is such that its rows sum to the null vector and all its square submatrices are invertible. □

6 Analytic construction for order up to 3

In this section, we develop explicit polynomial conditions on the entries of generalized Cauchy matrices that are sufficient to ensure both the correctness and safety of the two masking schemes described in Section 3.

The results are explicit constructions for many field sizes. For order $d = 1$, Corollary 15 proves that any non-zero $\boldsymbol{\gamma}$ matrix makes the scheme secure. For order $d = 2$, Corollary 16 proves that our MDS preconditions of the previous section always produce safe constructions without the need for any further checks. Finally, for order $d = 3$, Theorems 19 and 21 provide x_i and y_i values to use in order to generate safe Cauchy matrices for any field of characteristic 2 with $q \geq 4$.

The idea behind our preconditions in Section 5 was to ensure that all square submatrices of $\boldsymbol{\gamma}$ are non-singular, and therefore *many* square submatrices of the matrix $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ have nullity zero. For small dimensions, we can go further and actually require that *all* submatrices of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ which could possibly violate the condition $\mathcal{C}''$ from (5) are non-singular. This will in turn guarantee a safe and correct construction by Theorem 10 and Propositions 6 and 7.

6.1 Columns which must be selected

Let $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times n}$ and recall the definitions of $\boldsymbol{\Delta}(\boldsymbol{N}_n)$ and $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$; in the former case we show only the positions of the non-zero entries, which are the same whether $\boldsymbol{N}_n = \boldsymbol{L}_n$ or $\boldsymbol{N}_n = \boldsymbol{L}'_n$.

$$\boldsymbol{\Delta}(\boldsymbol{N}_n) = \begin{pmatrix} * & * & \cdots & * & * & * & \cdots & * & & * & * & \cdots & * \\ & * & \cdots & * & & * & \cdots & * & \cdots & & * & \cdots & * \\ & & \ddots & \vdots & & & \ddots & \vdots & & & & \ddots & \vdots \\ & & & * & & & & * & & & & & * \end{pmatrix},$$

$$\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}) = \begin{pmatrix} \gamma_{0,1} & \gamma_{0,1} & \cdots & \gamma_{0,1} & \gamma_{1,1} & \gamma_{1,1} & \cdots & \gamma_{1,1} & & \gamma_{d,1} & \gamma_{d,1} & \cdots & \gamma_{d,1} \\ & \gamma_{0,2} & \cdots & \gamma_{0,2} & & \gamma_{1,2} & \cdots & \gamma_{1,2} & \cdots & & \gamma_{d,2} & \cdots & \gamma_{d,2} \\ & & \ddots & \vdots & & & \ddots & \vdots & & & & \ddots & \vdots \\ & & & \gamma_{0,n} & & & & \gamma_{1,n} & & & & & \gamma_{d,n} \end{pmatrix}.$$

Notice that all pairs of columns in $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_n$ with the same index (hence corresponding to the same probe in the masking scheme) have the same weight. The next lemma shows that any unsafe set of probes from among these columns must include at least two of the full-weight columns.

Lemma 14. *Let $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times n}, \boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{L}_n$ be as above. If $\boldsymbol{\gamma}$ has no zero entries, then any column selection $\boldsymbol{P} \in S_n^{\ell_n}$ which is a counterexample to $\mathcal{C}'(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_n))$ must include at least two columns of full weight n from $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ and $\boldsymbol{\Delta}(\boldsymbol{N}_n)$.*

Proof. A counterexample to $\mathcal{C}'(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_n))$ is a selection matrix $\boldsymbol{P} \in S_n^{\ell_n}$ such that the matrix product $\boldsymbol{\Delta}(\boldsymbol{N}_n)\boldsymbol{P} \cdot \mathrm{kerb}(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})\boldsymbol{P})$ has no zero rows.

The only columns of $\boldsymbol{\Delta}(\boldsymbol{N}_n)$ which are non-zero in the last row are those columns of full weight, so at least one must be included in $\boldsymbol{P}$ for the product to have no zero rows. But in order for $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})\boldsymbol{P}$ to have a non-trivial kernel, it must have a *second* column with a non-zero in the last row. □

6.2 Dimensions 1 and 2

Combined with the results of the prior sections, this leads immediately to solutions for orders $n = 1$ or $n = 2$.

Corollary 15. *For any $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times 1}$ that contains no zero entries, we have $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_1)$.*

Proof. Clearly there is no way to include two full-weight columns in a selection $\boldsymbol{P} \in S_1^{\ell_1}$ of a single column. Therefore from Lemma 14, we have $\neg\mathcal{C}'(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_1))$. By Theorems 8 and 10 this implies the statement above. □

Corollary 16. *For any $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times 2}$ such that all square submatrices of $\boldsymbol{\gamma}$ are MDS, we have $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_2)$.*

Proof. Any selection of 2 columns of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ that includes at least 2 full-weight columns is simply a transposed submatrix of $\boldsymbol{\gamma}$ of dimension 2. By Theorem 1, any such submatrix is non-singular, and thus has a trivial kernel. Therefore by Lemma 14 there are no counterexamples to $\mathcal{C}'(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_2))$, and by Theorems 8 and 10 again the stated result follows. □

Most notably, these corollaries guarantee that *any* matrix with column dimension 1 or 2 which satisfies Precondition 4.1 or Precondition 5.1 is an instantiation of the respective masking scheme that is correct and safe. Because we have explicit constructions for these preconditions in Theorems 12 and 13 over any field $\mathbb{F}_q$ with $q > 2d+1$, we also have explicit instantiations for the masking schemes secure against 1 or 2 probes.

6.3 Dimension 3

Next we turn to the case of $n = 3$. It is no longer possible to construct safe instances of $\boldsymbol{\gamma}$ based on the MDS preconditions alone, but there is only one other shape of square submatrices that need be considered.

Lemma 17. *Let $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times 3}, \boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{L}_n$ be as above. If every square submatrix of $\boldsymbol{\gamma}$ is MDS, and for all distinct triples of indices $\{i, j, k\} \subseteq \{0, 1, \ldots, d+1\}$ the matrix*

$$\begin{pmatrix} \gamma_{i,1} & \gamma_{j,1} & \gamma_{k,1} \\ \gamma_{i,2} & \gamma_{j,2} & \gamma_{k,2} \\ \gamma_{i,3} & \gamma_{j,3} & 0 \end{pmatrix}$$

is non-singular, then we have $\mathcal{C}(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_3)$.

Proof. The goal is to ensure that no square submatrix of $\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}})$ which could possibly be part of a counterexample to $\mathcal{C}'(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_3))$ has a non-trivial kernel. Already we know from Lemma 14 that any such submatrix must include two distinct full-weight columns. Because all square submatrices of $\boldsymbol{\gamma}$ are MDS, these two columns have a trivial kernel, meaning a third column must be added if one hopes to find a counterexample. This leads to three cases, depending on the weight of this third column.

If the third column has weight 1, the situation is analogous to that of Example 11. The corresponding matrix is non-singular if and only if some 2×2 submatrix of $\boldsymbol{\gamma}$ is non-singular, which it must be by the MDS assumption.

Next, if the third column has full weight 3, then we have a 3×3 submatrix of $\boldsymbol{\gamma}$, which again must be non-singular.

The remaining case is that the third column has weight 2, as in the statement of the lemma. All that remains is to prove that this index k must be distinct from i and j. By way of contradiction, and without loss of generality, suppose $i = k$. Then after subtracting the third column from the first, we obtain the matrix

$$\begin{pmatrix} 0 & \gamma_{j,1} & \gamma_{i,1} \\ 0 & \gamma_{j,2} & \gamma_{i,2} \\ \gamma_{i,3} & \gamma_{j,3} & 0 \end{pmatrix},$$

which is non-singular if and only if the original matrix is non-singular. And indeed, this matrix must be non-singular because the upper-right 2×2 matrix is a submatrix of $\boldsymbol{\gamma}$.

Therefore the only remaining case of a submatrix which could be a counterexample to $\mathcal{C}'(\boldsymbol{\Delta}(\boldsymbol{M}'_{\boldsymbol{\gamma}}), \boldsymbol{\Delta}(\boldsymbol{N}_3))$ is one of the form given in the statement of the lemma. Applying once again Theorems 8 and 10 completes the proof. □

This finally leads to a way to construct safe instances for the schemes when $d = 3$ based only on polynomial conditions, via the following steps:

1. Write down a symbolic 4×3 matrix $\boldsymbol{\gamma}$ satisfying Precondition 4.1 or Precondition 5.1 according to the constructions of Theorem 12 or Theorem 13, leaving all the x_i's and y_i's as indeterminates.
2. Extract all 3×3 matrices from $\boldsymbol{\gamma}$ that match the form of Lemma 17 and compute their determinants, which are rational functions in the x_is and y_is.
3. Factor the numerators of all determinants, removing duplicate factors and factors such as $x_i - y_i$ which must be non-zero by construction.
4. A common non-root to the resulting list of polynomials corresponds to a $\boldsymbol{\gamma}$ matrix which is safe for the given scheme.

Next we show the results of these computations for each of the two schemes. We used the Sage [Sag16] computer algebra system to compute the lists of polynomials according to the procedure above, which takes about 1 second on a modern laptop computer.

Proposition 18. *If $x_1, x_2, x_3, y_1, y_2, y_3 \in \mathbb{F}_q$ are distinct non-zero elements so that the list of polynomials in Figure 1 all evaluate to non-zero values, then the*

$$
\begin{gathered}
x_2x_3 - y_1y_2 - x_2y_3 - x_3y_3 + y_1y_3 + y_2y_3 \\
x_2x_3 - x_3y_1 - x_3y_2 + y_1y_2 - x_2y_3 + x_3y_3 \\
x_2x_3 - x_2y_1 - x_2y_2 + y_1y_2 + x_2y_3 - x_3y_3 \\
x_1x_3 - y_1y_2 - x_1y_3 - x_3y_3 + y_1y_3 + y_2y_3 \\
x_1x_3 - x_3y_1 - x_3y_2 + y_1y_2 - x_1y_3 + x_3y_3 \\
x_1x_3 - x_1y_1 - x_1y_2 + y_1y_2 + x_1y_3 - x_3y_3 \\
x_1x_2 - y_1y_2 - x_1y_3 - x_2y_3 + y_1y_3 + y_2y_3 \\
x_1x_2 - x_2y_1 - x_2y_2 + y_1y_2 - x_1y_3 + x_2y_3 \\
x_1x_2 - x_1y_1 - x_1y_2 + y_1y_2 + x_1y_3 - x_2y_3 \\
x_2y_1y_2 - x_3y_1y_2 - x_2x_3y_3 + x_3y_1y_3 + x_3y_2y_3 - y_1y_2y_3 \\
x_2y_1y_2 - x_3y_1y_2 + x_2x_3y_3 - x_2y_1y_3 - x_2y_2y_3 + y_1y_2y_3 \\
x_1y_1y_2 - x_3y_1y_2 - x_1x_3y_3 + x_3y_1y_3 + x_3y_2y_3 - y_1y_2y_3 \\
x_1y_1y_2 - x_3y_1y_2 + x_1x_3y_3 - x_1y_1y_3 - x_1y_2y_3 + y_1y_2y_3 \\
x_1y_1y_2 - x_2y_1y_2 - x_1x_2y_3 + x_2y_1y_3 + x_2y_2y_3 - y_1y_2y_3 \\
x_1y_1y_2 - x_2y_1y_2 + x_1x_2y_3 - x_1y_1y_3 - x_1y_2y_3 + y_1y_2y_3 \\
x_2x_3y_1 + x_2x_3y_2 - x_2y_1y_2 - x_3y_1y_2 - x_2x_3y_3 + y_1y_2y_3 \\
x_1x_3y_1 + x_1x_3y_2 - x_1y_1y_2 - x_3y_1y_2 - x_1x_3y_3 + y_1y_2y_3 \\
x_1x_2y_1 + x_1x_2y_2 - x_1y_1y_2 - x_2y_1y_2 - x_1x_2y_3 + y_1y_2y_3 \\
x_1x_2x_3 - x_2x_3y_1 - x_2x_3y_2 - x_1y_1y_2 + x_2y_1y_2 + x_3y_1y_2 - x_1x_2y_3 - x_1x_3y_3 + x_2x_3y_3 + x_1y_1y_3 + x_1y_2y_3 - y_1y_2y_3 \\
x_1x_2x_3 - x_1x_3y_1 - x_1x_3y_2 + x_1y_1y_2 - x_2y_1y_2 + x_3y_1y_2 - x_1x_2y_3 + x_1x_3y_3 - x_2x_3y_3 + x_2y_1y_3 + x_2y_2y_3 - y_1y_2y_3 \\
x_1x_2x_3 - x_1x_2y_1 - x_1x_2y_2 + x_1y_1y_2 + x_2y_1y_2 - x_3y_1y_2 + x_1x_2y_3 - x_1x_3y_3 - x_2x_3y_3 + x_3y_1y_3 + x_3y_2y_3 - y_1y_2y_3
\end{gathered}
$$

Fig. 1: Polynomials which should be non-zero to generate a safe construction according to Condition 4.1. There are 9 degree-2 polynomials with 6 terms, 9 degree-3 polynomials with 6 terms, and 3 degree-3 polynomials with 12 terms.

matrix $\boldsymbol{\gamma}$ constructed according to Theorem 12 generates a safe masking scheme according to Condition 4.1.

From the degrees of these polynomials, and by the Schwartz-Zippel lemma [Sch80] and applying the union bound, a safe construction for Condition 4.1 exists over any field $\mathbb{F}_q$ with $q > 54$.

In fact, we have an explicit construction for any binary field $\mathbb{F}_q$ with $q \geq 16$.

Theorem 19. *Let $(x_1, x_2, x_3) = (1, 3, 5)$ and $(y_1, y_2, y_3) = (6, 4, a)$. Then for any $k \geq 4$, the matrix $\boldsymbol{\gamma}$ constructed according to Theorem 12 generates a safe masking scheme over $\mathbb{F}_{2^k}$ according to Condition 4.1.*

Proof. Small cases with $4 \leq k \leq 8$ are checked computationally by making the appropriate substitutions into the polynomials of Figure 1.

For $k \geq 9$, consider the degrees of the x_is and y_is when treated as polynomials over $\mathbb{F}_2$. The highest degree is $\deg y_3 = 3$, and all other elements have degree at most 2. Inspecting the polynomials in Figure 1, we see that they are all sums of products of at most three distinct variables. Therefore, when evaluated at these x_is and y_is, the degree of any resulting polynomial is at most 7. Over $\mathbb{F}_{2^k}$ where $k \geq 8$ there is therefore no reduction, and the polynomials are guaranteed to be non-zero in all cases because they are non-zero over $\mathbb{F}_{2^8}$. □

Next we do the same for the masking scheme with linear randomness complexity, namely that of Condition 5.1.

$$x_2x_3x_4 - x_3x_4y_1 - x_3x_4y_2 - x_2y_1y_2 + x_3y_1y_2 + x_4y_1y_2 - x_2x_3y_3 - x_2x_4y_3 + x_3x_4y_3 + x_2y_1y_3 + x_2y_2y_3 - y_1y_2y_3$$
$$x_2x_3x_4 - x_2x_4y_1 - x_2x_4y_2 + x_2y_1y_2 - x_3y_1y_2 + x_4y_1y_2 - x_2x_3y_3 + x_2x_4y_3 - x_3x_4y_3 + x_3y_1y_3 + x_3y_2y_3 - y_1y_2y_3$$
$$x_2x_3x_4 - x_2x_3y_1 - x_2x_3y_2 + x_2y_1y_2 + x_3y_1y_2 - x_4y_1y_2 + x_2x_3y_3 - x_2x_4y_3 - x_3x_4y_3 + x_4y_1y_3 + x_4y_2y_3 - y_1y_2y_3$$
$$x_1x_3x_4 - x_3x_4y_1 - x_3x_4y_2 - x_1y_1y_2 + x_3y_1y_2 + x_4y_1y_2 - x_1x_3y_3 - x_1x_4y_3 + x_3x_4y_3 + x_1y_1y_3 + x_1y_2y_3 - y_1y_2y_3$$
$$x_1x_3x_4 - x_1x_4y_1 - x_1x_4y_2 + x_1y_1y_2 - x_3y_1y_2 + x_4y_1y_2 - x_1x_3y_3 + x_1x_4y_3 - x_3x_4y_3 + x_3y_1y_3 + x_3y_2y_3 - y_1y_2y_3$$
$$x_1x_3x_4 - x_1x_3y_1 - x_1x_3y_2 + x_1y_1y_2 + x_3y_1y_2 - x_4y_1y_2 + x_1x_3y_3 - x_1x_4y_3 - x_3x_4y_3 + x_4y_1y_3 + x_4y_2y_3 - y_1y_2y_3$$
$$x_1x_2x_4 - x_2x_4y_1 - x_2x_4y_2 - x_1y_1y_2 + x_2y_1y_2 + x_4y_1y_2 - x_1x_2y_3 - x_1x_4y_3 + x_2x_4y_3 + x_1y_1y_3 + x_1y_2y_3 - y_1y_2y_3$$
$$x_1x_2x_4 - x_1x_4y_1 - x_1x_4y_2 + x_1y_1y_2 - x_2y_1y_2 + x_4y_1y_2 - x_1x_2y_3 + x_1x_4y_3 - x_2x_4y_3 + x_2y_1y_3 + x_2y_2y_3 - y_1y_2y_3$$
$$x_1x_2x_4 - x_1x_2y_1 - x_1x_2y_2 + x_1y_1y_2 + x_2y_1y_2 - x_4y_1y_2 + x_1x_2y_3 - x_1x_4y_3 - x_2x_4y_3 + x_4y_1y_3 + x_4y_2y_3 - y_1y_2y_3$$
$$x_1x_2x_3 - x_2x_3y_1 - x_2x_3y_2 - x_1y_1y_2 + x_2y_1y_2 + x_3y_1y_2 - x_1x_2y_3 - x_1x_3y_3 + x_2x_3y_3 + x_1y_1y_3 + x_1y_2y_3 - y_1y_2y_3$$
$$x_1x_2x_3 - x_1x_3y_1 - x_1x_3y_2 + x_1y_1y_2 - x_2y_1y_2 + x_3y_1y_2 - x_1x_2y_3 + x_1x_3y_3 - x_2x_3y_3 + x_2y_1y_3 + x_2y_2y_3 - y_1y_2y_3$$
$$x_1x_2x_3 - x_1x_2y_1 - x_1x_2y_2 + x_1y_1y_2 + x_2y_1y_2 - x_3y_1y_2 + x_1x_2y_3 - x_1x_3y_3 - x_2x_3y_3 + x_3y_1y_3 + x_3y_2y_3 - y_1y_2y_3$$

Fig. 2: Polynomials which should be non-zero to generate a safe construction according to Condition 5.1. There are 12 degree-3 polynomials with 12 terms each.

Proposition 20. *If $x_1, x_2, x_3, x_4, y_1, y_2, y_3 \in \mathbb{F}_q$ are distinct non-zero elements so that the list of polynomials in Figure 2 all evaluate to non-zero values, then the matrix constructed according to Theorem 13 generates a safe masking scheme according to Condition 5.1.*

Applying the Schwartz-Zippel lemma and union bound in this context guarantees a safe construction for Condition 5.1 over any field $\mathbb{F}_q$ with $q > 36$. Again, we have an explicit construction for binary fields of order at least 16.

Theorem 21. *Let $(x_1, x_2, x_3, x_4) = (1, 2, 5, 6)$ and $(y_1, y_2, y_3) = (4, 7, f)$. Then for any $k \geq 4$, the matrix $\boldsymbol{\gamma}$ constructed according to Theorem 13 generates a safe masking scheme over $\mathbb{F}_{2^k}$ according to Condition 5.1.*

The proof is the same as Theorem 19, consisting of computational checks for $4 \leq k \leq 8$ and then an argument for all $k \geq 9$ based on the degrees of the x_i and y_i polynomials.

7 Efficient algorithms to test safeness

We now turn to a computational approach, in order to deal with the schemes at order $d > 3$ that were not treated in the previous section.

To test whether a matrix may be used to safely instantiate either of the masking schemes of Belaïd *et al.*, we use the condition $\mathcal{C}'(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_d)$ defined in (4), which according to Theorem 8 is a sufficient condition for the scheme under consideration to be safe. The definition of this condition immediately indicates an algorithm, which we have implemented with some optimizations, using M4RIE [Alb13] for the finite field arithmetic.

7.1 The algorithm

To test whether a matrix $\boldsymbol{\gamma} \in \mathbb{K}^{(d+1)\times d}$ satisfies the conditions of Proposition 6 or Proposition 7, simply construct $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_d$ and for all d-subsets of columns $\boldsymbol{P} \in S_d^\ell$, check if $\mathcal{Z}(\boldsymbol{N}_d\boldsymbol{P} \cdot \mathrm{kerb}(\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{P}))$.

This algorithm is much more efficient than the one directly suggested by Condition 4.1: instead of testing all $\sum_{i=1}^{d} \binom{\ell}{i} q^i$ vectors of $\mathbb{F}_q^\ell$ of weight d or less, it is enough to do $\binom{\ell}{d}$ easy linear algebra computations. While this remains exponential in d, it removes the practically insuperable factor q^d and gives a complexity that does not depend on the field size (save for the cost of arithmetic).

(Note that we could have used the condition $\mathcal{C}''$ as in Theorem 10 instead, but this turns out to be more complicated in practice due to the need to take arbitrary subsets of the rows and columns of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ and $\boldsymbol{N}_d$.)

We now describe two implementation strategies for this algorithm.

7.2 Straightforward implementation with optimizations

Two simple optimizations may be used to make a straightforward implementation of the above algorithm more efficient in practice.

Skipping bad column picks. We can see already from the support of $\boldsymbol{N}_d$ that some subsets of columns $\boldsymbol{P} \in S_d^\ell$ never need to be checked because $\mathcal{Z}(\boldsymbol{N}_d\boldsymbol{P})$ is already true, independent of the actual choice of $\boldsymbol{\gamma}$. This is the case for example when the columns selected by $\boldsymbol{P}$ are all of weight 1.

For the specific cases of $d = 4$, this reduces the number of supports to be considered from $\binom{49}{4} = 211\,876$ to $103\,030$, saving roughly a factor 2. A similar behaviour is observed for $d = 5$, when one only has to consider $6\,448\,239$ supports among the $\binom{71}{5} = 13\,019\,909$ possible ones. Note that the same optimization could be applied to the naïve algorithm that exhaustively enumerates low-weight vectors of $\mathbb{F}_q^\ell$.

Testing critical cases first. Looking again at how $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ is defined, it is easy to see that for some column selections $\boldsymbol{P}$, $\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{P}$ does not in fact depend on $\boldsymbol{\gamma}$. For these, it is enough to check once and for all that $\mathcal{Z}(\boldsymbol{N}_{\boldsymbol{\gamma}}\boldsymbol{P} \cdot \mathrm{kerb}(\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{P}))$ indeed holds (if it does not, the scheme would be generically broken). Going further, even some column subsets such that $\boldsymbol{M}_{\boldsymbol{\gamma}}\boldsymbol{P}$ actually depends on $\boldsymbol{\gamma}$ may always be "safe" provided that $\boldsymbol{\gamma}$ satisfies a certain precondition, such as for instance being MDS, as suggested in Section 5.

Conversely, it may be the case that for some $\boldsymbol{P}$, $\mathcal{Z}(\boldsymbol{N}_d\boldsymbol{P} \cdot \mathrm{kerb}(\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{P}))$ often does *not* hold. It may then be beneficial to test this subset $\boldsymbol{P}$ before others that are less likely to make the condition fail. We have experimentally observed that such subsets do exist. For instance, in the case $d = 5$ for Condition 4.1, only $\approx 320\,000$ column subsets seem to determine whether a matrix satisfies the condition or not.[¶] There, checking these supports first and using an early-abort

[¶]This figure was found experimentally by regrouping the supports in clusters of 10 000, independently of q. A more careful analysis may lead to a more precise result.

strategy, verifying that a matrix *does not* satisfy the condition is at least ≈ 20 times faster than enumerating all possible column subsets.

7.3 Batch implementation

Especially when the matrix $\boldsymbol{\gamma}$ under consideration actually satisfies the required conditions, checking these using the straightforward strategy entails considerable redundant computation due to the overlap between subsets of columns.

To avoid this, we also implemented a way to check the condition $\mathcal{C}'(\boldsymbol{M}'_{\boldsymbol{\gamma}}, \boldsymbol{N}_d)$ that operates over the entire matrix simultaneously, effectively considering many subsets of columns in a single batch.

Recall that the algorithm needs to (1) extract a subset of columns of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$, (2) compute a right kernel basis for this subset, (3) multiply $\boldsymbol{N}_d$ times this kernel basis, and (4) check for zero rows in the resulting product.

Steps (2) and (3) would typically be performed via Gaussian elimination: For each column of $\boldsymbol{M}'_{\boldsymbol{\gamma}}$ that is in the selection, we search for a pivot row, permute rows if necessary to move the pivot up, then eliminate above and below the pivot and move on. If there is no pivot in some column, this means a new null vector has been found; we use the previous pivots to compute the null vector and add it to the basis. Finally, we multiply this null space basis by the corresponding columns in $\boldsymbol{N}_d$ and check for zero rows.

The key observation for this algorithm is that we can perform these steps (2) and (3) *in parallel* to add one more column to an existing column selection. That is, starting with some subset of columns, we consider the effect on the null space basis and the following multiplication by $\boldsymbol{N}_d$ simultaneously for all other columns in the matrices. Adding columns with pivots does not change the null space basis or the product with $\boldsymbol{N}_d$. Columns with no pivots add one additional column to the null space basis, which results in a new column in the product with $\boldsymbol{N}_d$. This new column of $\boldsymbol{N}_d\boldsymbol{P} \cdot \ker(\boldsymbol{M}'_{\boldsymbol{\gamma}}\boldsymbol{P})$ may be checked for non-zero entries and then immediately discarded as the search continues; in later steps, the *rows* of this product which already have a non-zero entry no longer need to be considered.

All of this effectively reduces the cost of the check by a factor of ℓ compared to the prior version, replacing the search over all size-d subsets with a search over size-$(d-1)$ subsets and some matrix computations. This strategy is especially effective when the $\boldsymbol{\gamma}$ matrix under consideration is (nearly or actually) safe, meaning that the early termination techniques above will not be very useful.

8 Experimental results and explicit instantiations

We implemented both algorithms of the previous section in the practically-useful case of binary fields, using M4RIE for the underlying linear algebra [Alb13], and searched for matrices fulfilling Conditions 4.1 and 5.1 in various settings, leading to instantiations of the masking schemes of Belaïd *et al.* up to $d = 6$

and $\mathbb{F}_{2^{16}}$.[‖] We also collected statistics about the fraction of matrices satisfying the conditions, notably in function of the field over which they are defined, and experimentally verified the usefulness of Precondition 4.1.

8.1 Statistics

We give detailed statistics about the proportion of preconditioned matrices allowing to instantiate either masking scheme up to order 6; this is presented in Tables 1 and 2. The data was collected by drawing at random matrices satisfying Precondition 4.1 or Precondition 5.1 and checking if they satisfied the safety conditions or not for the respective scheme.

For combinations of field size and order where no safe matrix was found, we give the result as an upper bound.

Notice that the probability for Condition 5.1 appears to be consistently a bit higher than that for Condition 4.1. The combinations of field size q and order d where safe instances are found were almost the same for both schemes, except for order 5 and $q = 2^9$, where a safe preconditioned matrix was found for Condition 5.1 but not for Condition 4.1. This difference between the schemes may be explained by the fact that Condition 4.1 places conditions on two matrices $\boldsymbol{\gamma}$ and $\mathbf{1}_{d\times d} - \boldsymbol{\gamma}$, whereas Condition 5.1 depends only on the single matrix $\boldsymbol{\gamma}$.

An important remark is that for the smallest field $\mathbb{F}_{2^5}$, the statistics do not include results about the *non-preconditioned safe matrices*, which were the only safe ones we found, see the further discussion below.

We indicate the sample sizes used to obtain each result, as they may vary by several orders of magnitude due to the exponentially-increasing cost of our algorithm with the order. As an illustration, our batch implementation is able to check 1 000 000 dimension-4 matrices over $\mathbb{F}_{2^6}$ in 12 400 seconds on one core of a 2 GHz Sandy Bridge CPU, which increases to 590 000 and 740 000 seconds for $\mathbb{F}_{2^{12}}$ and $\mathbb{F}_{2^{16}}$ respectively because of more expensive field operations; 1 600 000 seconds allowed to test $\approx$ 145 000 and $\approx$ 25 000 dimension-5 matrices for these last two fields, and $\approx$ 2 400 dimension-6 matrices for $\mathbb{F}_{2^{16}}$.

Usefulness of the preconditions. We now address the question of the usefulness of the preconditions of Section 5. Our goal is to determine with what probability randomly-generated matrices in fact already satisfy the preconditions, and whether doing so for a matrix $\boldsymbol{\gamma}$ has a positive impact on its satisfying Condition 4.1 or Condition 5.1.

We did this experimentally in two settings, both for the first scheme corresponding to Condition 4.1: order $d = 4$ over $\mathbb{F}_{2^8}$ and order $d = 5$ over $\mathbb{F}_{2^{13}}$. We generated enough random matrices $\boldsymbol{\gamma}$ in order to obtain respectively 20 000 and 2 000 of them satisfying Condition 4.1, and counted how many of the corresponding safe pairs $(\boldsymbol{\gamma}, \mathbf{1}_{d\times d} - \boldsymbol{\gamma})$ had at least one or both elements that were MDS and XMDS. The same statistics were gathered for all the generated matrices,

[‖] $\mathbb{F}_{2^{16}}$ is the largest field size implemented in M4RIE, and $d = 6$ the maximum dimension for which safe instantiations (seem to) exist below this field size limitation.

Table 1: Instantiations over $\mathbb{F}_{2^5} \sim \mathbb{F}_{2^{10}}$. Sample sizes (as indicated by symbols in the exponents) were as follows: $* \approx 400\,000$; $\ddagger = 1\,000\,000$; $\star \approx 4\,000\,000$; $\dagger \approx 11\,000\,000$.

q	2^5	2^6	2^7	2^8	2^9	2^{10}
d			Condition 4.1 & Precondition 4.1			
4	$\leq 2^{-28.8}$	$2^{-15.25\dagger}$	$0.009^{\dagger}$	$0.11^{\ddagger}$	$0.34^{\ddagger}$	$0.59^{\ddagger}$
5	—	—	—	—	$\leq 2^{-27.5}$	$2^{-18.9\star}$
d			Condition 5.1 & Precondition 5.1			
4	$\leq 2^{-33.5}$	$2^{-9.10\ddagger}$	$0.062^{\ddagger}$	$0.27^{\ddagger}$	$0.53^{\ddagger}$	$0.73^{\ddagger}$
5	—	—	—	—	$2^{-18.6*}$	$2^{-11.0*}$

Table 2: Instantiations over $\mathbb{F}_{2^{11}} \sim \mathbb{F}_{2^{16}}$. Sample sizes (as indicated by symbols in the exponents) were as follows: $\ddagger = 1\,000\,000$; $* \approx 400\,000$; $\diamond \approx 145\,000$; $\bullet \approx 65\,000$; $\triangleleft \approx 40\,000$; $\oslash \approx 30\,000$; $\ltimes \approx 25\,000$; $\wr \approx 560\,000$; $\curlywedge \approx 12\,700$.

q	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
d			Condition 4.1 & Precondition 4.1			
4	$0.77^{\ddagger}$	$0.88^{\ddagger}$	$0.94^{\ddagger}$	$0.97^{\ddagger}$	$0.98^{\ddagger}$	$0.99^{\ddagger}$
5	0.0015^{*}	$0.04^{\diamond}$	$0.2^{\bullet}$	$0.45^{\triangleleft}$	$0.67^{\oslash}$	$0.82^{\ltimes}$
6	—	—	—	—	$2^{-16.8\wr}$	$0.003^{\curlywedge}$
d			Condition 5.1 & Precondition 5.1			
4	$0.86^{\ddagger}$	$0.92^{\ddagger}$	$0.96^{\ddagger}$	$0.98^{\ddagger}$	$0.99^{\ddagger}$	$1.00^{\ddagger}$
5	0.021^{*}	0.14^{*}	0.39^{*}	0.62^{*}	0.78^{*}	0.89^{*}
6	—	—	—	—	$2^{-12.7\triangleleft}$	$0.002^{\triangleleft}$

including the ones that were not safe. The results are respectively summarized in Tables 3 and 4.

Table 3: Case $d = 4$ over $\mathbb{F}_{2^8}$, for Condition 4.1.

	Total	One+ MDS	Both MDS	One+ XMDS	Both XMDS
#Random	672 625	634 096	389 504	515 840	315 273
#Safe	20 000	19 981	19 981	19 981	19 981
Ratio	0.030	0.032	0.051	0.039	0.063

A first comment on the results is that as already remarked in Section 5, the preconditions are not necessary to find safe instantiations. Indeed, for a few of the smallest cases $d = 3, q = 2^3$ and $d = 4, q = 2^5$, we were only able to find safe instantiations that did *not* meet the preconditions. For example, one can clearly see that the leading 2×2 submatrix of the following matrix is singular,

Table 4: Case $d = 5$ over $\mathbb{F}_{2^{13}}$, for Condition 4.1.

	Total	One+ MDS	Both MDS	One+ XMDS	Both XMDS
#Random	15 877	15 867	14 978	15 486	14 623
#Safe	2 000	2 000	2 000	2 000	2 000
Ratio	0.13	0.13	0.13	0.13	0.14

and hence the matrix is not MDS:

$$\boldsymbol{\gamma} = \begin{pmatrix} 4 & 2 & 6 \\ 4 & 2 & 3 \\ 4 & 2 & 3 \end{pmatrix}.$$

Yet (surprisingly), $\boldsymbol{\gamma}$ and $\mathbf{1} - \boldsymbol{\gamma}$ satisfy all requirements of Condition 4.1 over $\mathbb{F}_{2^3}$.

Nonetheless, the precondition is clearly helpful in the vast majority of cases. From our experiments, *in cases where any preconditioned safe matrix exists*, then nearly all safe matrices satisfy the precondition, while a significant fraction of random matrices do not. Enforcing the precondition by construction or as a first check is then indeed a way to improve the performance of a random search of a safe matrix. This is especially true for larger orders; for example, we did not find any safe matrices for order $d = 6$ over $\mathbb{F}_{2^{15}}$ by random search, but only by imposing Precondition 4.1.

Lastly, one should notice that specifically considering Cauchy matrices seems to further increase the odds of a matrix being safe, beyond the fact that it satisfies Condition 4.1: in the case $d = 4$, $\mathbb{F}_{2^8}$, Table 1 gives a success probability of 0.11, which is significantly larger than the 0.063 of Table 3, and in the case $d = 5$, $\mathbb{F}_{2^{13}}$, Table 2 gives 0.2, also quite higher than the 0.14 of Table 4. As of yet, we do not have an explanation for this observation.

8.2 Instantiations of [BBP+17, §4]

We conclude by giving explicit matrices allowing to safely instantiate the scheme of [BBP+17, §4] over various binary fields from order 3 up to 6; the case of order at most 2 is treated in Section 6 (Belaïd *et al.* also provided examples for $d = 2$). Our examples include practically-relevant instances with $d = 3, 4$ over $\mathbb{F}_{2^8}$.

We only give one matrix $\boldsymbol{\gamma}$ for every case we list, but we emphasise that as is required by the masking scheme, this means that both $\boldsymbol{\gamma}$ and $\boldsymbol{\delta} = \mathbf{1}_{d\times d} - \boldsymbol{\gamma}$ satisfy Condition 4.1. We list instances only for the smallest field size we know of, and for $\mathbb{F}_{2^8}$ (when applicable), but have computed explicit instances for all field sizes up to $\mathbb{F}_{2^{16}}$. These are given in the full version of this paper [KR18, App. A].

Instantiations at order 3. The smallest field for which we could find an instantiation at order 3 was $\mathbb{F}_{2^3}$. Recall that we also have an explicit construction in Section 6 for any 2^k with $k \geq 4$.

$$\gamma(\mathbb{F}_{2^3}) = \begin{pmatrix} 3 & 5 & 4 \\ 3 & 6 & 7 \\ 3 & 5 & 4 \end{pmatrix} \qquad \gamma(\mathbb{F}_{2^8}) = \begin{pmatrix} \texttt{e3} & \texttt{b7} & \texttt{50} \\ \texttt{bd} & \texttt{e8} & \texttt{8b} \\ \texttt{53} & \texttt{25} & \texttt{a0} \end{pmatrix}$$

Instantiations at order 4. The smallest field for which we could find an instantiation at order 4 was $\mathbb{F}_{2^5}$. The following matrices $\gamma(\mathbb{F}_q)$ may be used to instantiate the scheme over $\mathbb{F}_q$.

$$\gamma(\mathbb{F}_{2^5}) = \begin{pmatrix} \texttt{1c} & \texttt{c} & \texttt{1e} & \texttt{b} \\ \texttt{1c} & \texttt{c} & \texttt{1e} & \texttt{12} \\ \texttt{10} & \texttt{18} & \texttt{17} & \texttt{14} \\ \texttt{1c} & \texttt{c} & \texttt{1e} & \texttt{10} \end{pmatrix} \gamma(\mathbb{F}_{2^8}) = \begin{pmatrix} \texttt{56} & \texttt{5e} & \texttt{a1} & \texttt{3d} \\ \texttt{97} & \texttt{27} & \texttt{71} & \texttt{c7} \\ \texttt{f5} & \texttt{ae} & \texttt{68} & \texttt{88} \\ \texttt{1c} & \texttt{3} & \texttt{9c} & \texttt{8e} \end{pmatrix}$$

Instantiations at order 5. The smallest field for which we could find an instantiation at order 5 was $\mathbb{F}_{2^{10}}$. The following matrix may be used to instantiate the scheme over $\mathbb{F}_{2^{10}}$.

$$\gamma(\mathbb{F}_{2^{10}}) = \begin{pmatrix} \texttt{276} & \texttt{13e} & \texttt{64} & \texttt{1ab} & \texttt{120} \\ \texttt{189} & \texttt{181} & \texttt{195} & \texttt{30f} & \texttt{3fe} \\ \texttt{20a} & \texttt{3a1} & \texttt{199} & \texttt{30} & \texttt{2db} \\ \texttt{156} & \texttt{1ab} & \texttt{2f8} & \texttt{e5} & \texttt{2a8} \\ \texttt{303} & \texttt{321} & \texttt{265} & \texttt{d8} & \texttt{3a} \end{pmatrix}$$

Instantiations at order 6. The smallest field for which we could find an instantiation at order 6 was $\mathbb{F}_{2^{15}}$. The following matrix may be used to instantiate the scheme over $\mathbb{F}_{2^{15}}$.

$$\gamma(\mathbb{F}_{2^{15}}) = \begin{pmatrix} \texttt{151d} & \texttt{5895} & \texttt{5414} & \texttt{392b} & \texttt{2092} & \texttt{29a6} \\ \texttt{5c69} & \texttt{2f9e} & \texttt{241d} & \texttt{2ef7} & \texttt{baa} & \texttt{6f40} \\ \texttt{6e0d} & \texttt{8cf} & \texttt{7ca1} & \texttt{6503} & \texttt{23dc} & \texttt{6b3b} \\ \texttt{10d7} & \texttt{588e} & \texttt{2c22} & \texttt{1245} & \texttt{6a38} & \texttt{6484} \\ \texttt{1637} & \texttt{7062} & \texttt{2ae0} & \texttt{d1b} & \texttt{5305} & \texttt{381f} \\ \texttt{23f6} & \texttt{7d5} & \texttt{21bf} & \texttt{2879} & \texttt{2033} & \texttt{4377} \end{pmatrix}$$

8.3 Instantiations of [BBP+17, §5]

We now give similar instantiation results for the scheme with linear randomness complexity. This time, only a single matrix of dimension $(d+1) \times d$ is necessary to obtain a d-NI scheme. As in the previous case, we only focus here on the cases where $3 \leq d \leq 6$, and only list the matrices over the smallest binary field we have as well as $\mathbb{F}_{2^8}$ (where possible). We refer to [KR18] for all other cases.

Instantiations at order 3. The smallest field for which we could find an instantiation at order 3 was $\mathbb{F}_{2^3}$. Recall that we also have an explicit construction in Section 6 for any 2^k with $k \geq 4$.

$$\boldsymbol{\gamma}(\mathbb{F}_{2^3}) = \begin{pmatrix} 1 & 7 & 4 \\ 4 & 4 & 4 \\ 2 & 1 & 4 \\ 7 & 2 & 4 \end{pmatrix} \qquad \boldsymbol{\gamma}(\mathbb{F}_{2^8}) = \begin{pmatrix} \texttt{da} & \texttt{d5} & \texttt{e6} \\ \texttt{e8} & \texttt{1d} & \texttt{44} \\ \texttt{ad} & \texttt{b3} & \texttt{ce} \\ \texttt{9f} & \texttt{7b} & \texttt{6c} \end{pmatrix}$$

Instantiations at order 4. The smallest field for which we could find an instantiation at order 4 was $\mathbb{F}_{2^5}$. The following matrices $\gamma(\mathbb{F}_q)$ may be used to instantiate the scheme over $\mathbb{F}_q$.

$$\boldsymbol{\gamma}(\mathbb{F}_{2^5}) = \begin{pmatrix} \texttt{17} & \texttt{f} & \texttt{13} & \texttt{16} \\ \texttt{b} & \texttt{7} & \texttt{1a} & \texttt{11} \\ \texttt{1} & \texttt{1e} & \texttt{19} & \texttt{3} \\ \texttt{1b} & \texttt{10} & \texttt{2} & \texttt{a} \\ \texttt{6} & \texttt{6} & \texttt{12} & \texttt{e} \end{pmatrix} \boldsymbol{\gamma}(\mathbb{F}_{2^8}) = \begin{pmatrix} \texttt{ac} & \texttt{39} & \texttt{c0} & \texttt{36} \\ \texttt{79} & \texttt{5f} & \texttt{d9} & \texttt{51} \\ \texttt{9d} & \texttt{16} & \texttt{ca} & \texttt{63} \\ \texttt{a3} & \texttt{cb} & \texttt{6} & \texttt{81} \\ \texttt{eb} & \texttt{bb} & \texttt{d5} & \texttt{85} \end{pmatrix}$$

Instantiations at order 5. The smallest field for which we could find an instantiation at order 5 was $\mathbb{F}_{2^9}$. The following matrix may be used to instantiate the scheme over $\mathbb{F}_{2^9}$.

$$\boldsymbol{\gamma}(\mathbb{F}_{2^9}) = \begin{pmatrix} \texttt{7d} & \texttt{12c} & \texttt{18} & \texttt{1a3} & \texttt{da} \\ \texttt{121} & \texttt{131} & \texttt{109} & \texttt{1a7} & \texttt{3b} \\ \texttt{4a} & \texttt{131} & \texttt{91} & \texttt{a4} & \texttt{1c4} \\ \texttt{17c} & \texttt{cb} & \texttt{14b} & \texttt{41} & \texttt{57} \\ \texttt{fd} & \texttt{87} & \texttt{ac} & \texttt{17a} & \texttt{149} \\ \texttt{97} & \texttt{160} & \texttt{67} & \texttt{19b} & \texttt{3b} \end{pmatrix}$$

Instantiations at order 6. The smallest field for which we could find an instantiation at order 6 was $\mathbb{F}_{2^{15}}$. The following matrix may be used to instantiate the scheme over $\mathbb{F}_{2^{15}}$.

$$\boldsymbol{\gamma}(\mathbb{F}_{2^{15}}) = \begin{pmatrix} \texttt{475c} & \texttt{77e7} & \texttt{64ef} & \texttt{7893} & \texttt{4cd1} & \texttt{6e20} \\ \texttt{63dd} & \texttt{71f} & \texttt{29da} & \texttt{600e} & \texttt{36be} & \texttt{1db7} \\ \texttt{5511} & \texttt{d63} & \texttt{3719} & \texttt{4874} & \texttt{664} & \texttt{5014} \\ \texttt{410e} & \texttt{7cf2} & \texttt{9d9} & \texttt{10a1} & \texttt{7525} & \texttt{6098} \\ \texttt{7bfe} & \texttt{2998} & \texttt{7e20} & \texttt{1438} & \texttt{35e6} & \texttt{51e} \\ \texttt{7564} & \texttt{75d3} & \texttt{221a} & \texttt{67c7} & \texttt{56f1} & \texttt{18d5} \\ \texttt{3e04} & \texttt{5d22} & \texttt{2fcf} & \texttt{33b7} & \texttt{6a39} & \texttt{5ed0} \end{pmatrix}$$

8.4 Minimum field sizes for safe instantiations

We conclude by briefly comparing the minimum field sizes for which we could find safe instantiations of Condition 4.1 and Condition 5.1 with the ones given

by the non-constructive existence theorems of Belaïd *et al.*. Namely, [BBP+17, Thm. 4.5] guarantees the existence of a pair of safe matrices for Condition 4.1 in dimension d over $\mathbb{F}_q$ as long as $q > 2d \cdot (12d)^d$, and [BBP+17, Thm. 5.4] of a safe matrix for Condition 5.1 as long as $q > d \cdot (d+1) \cdot (12d)^d$. We give in Table 5 the explicit values provided by these two theorems for $2 \leq d \leq 6$ and q a power of two, along with the experimental minima that we found. From these, it seems that the sufficient condition of Belaïd *et al.* is in fact rather pessimistic.

Table 5: Sufficient field sizes for safe instantiations in characteristic two. Sizes are given as $\log(q)$.

d / $\min(\log(q))$	[BBP+17, Thm. 4.5]	Section 8.2	[BBP+17, Thm. 5.4]	Section 8.3
2	11	3	12	3
3	19	3	20	3
4	26	5	27	5
5	33	10	35	9
6	41	15	43	15

Acknowledgements

We thank Daniel Augot for the interesting discussions we had in the early stages of this work.

This work was performed while the second author was graciously hosted by the Laboratoire Jean Kuntzmann at the Université Grenoble Alpes.

The first author was supported in part by the French National Research Agency through the framework of the "Investissements d'avenir" program (ANR-15-IDEX-02).

The second author was supported in part by the National Science Foundation under grants #1319994 and #1618269, and in part by the Office of Naval Research award #N0001417WX01516.

Some of the computations were performed using the Grace supercomputer hosted by the U.S. Naval Academy Center for High Performance Computing, with funding from the DoD HPC Modernization Program.

References

Alb13. Martin Albrecht, *The M4RIE Library*, The M4RIE Team, 2013.

BBD+16. Gilles Barthe, Sonia Belaïd, François Dupressoir, Pierre-Alain Fouque, Benjamin Grégoire, Pierre-Yves Strub, and Rébecca Zucchini, *Strong Non-Interference and Type-Directed Higher-Order Masking*, ACM CCS 2016

(Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, eds.), ACM, 2016, pp. 116–129.

BBP+16. Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud, *Randomness Complexity of Private Circuits for Multiplication*, EUROCRYPT 2016 (Marc Fischlin and Jean-Sébastien Coron, eds.), Lecture Notes in Computer Science, vol. 9666, Springer, 2016, pp. 616–648.

BBP+17. Sonia Belaïd, Fabrice Benhamouda, Alain Passelègue, Emmanuel Prouff, Adrian Thillard, and Damien Vergnaud, *Private Multiplication over Finite Fields*, CRYPTO 2017 (Jonathan Katz and Hovav Shacham, eds.), Lecture Notes in Computer Science, vol. 10403, Springer, 2017, pp. 397–426.

CJRR99. Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi, *Towards Sound Approaches to Counteract Power-Analysis Attacks*, in Wiener [Wie99], pp. 398–412.

CPRR16. Claude Carlet, Emmanuel Prouff, Matthieu Rivain, and Thomas Roche, *Algebraic Decomposition for Probing Security*, IACR Cryptology ePrint Archive **2016** (2016), 321.

GP99. Louis Goubin and Jacques Patarin, *DES and Differential Power Analysis (The "Duplication" Method)*, CHES'99 (Çetin Kaya Koç and Christof Paar, eds.), Lecture Notes in Computer Science, vol. 1717, Springer, 1999, pp. 158–172.

ISW03. Yuval Ishai, Amit Sahai, and David A. Wagner, *Private Circuits: Securing Hardware against Probing Attacks*, CRYPTO 2003 (Dan Boneh, ed.), Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. 463–481.

KJJ99. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun, *Differential Power Analysis*, in Wiener [Wie99], pp. 388–397.

KR18. Pierre Karpman and Daniel S. Roche, *New Instantiations of the CRYPTO 2017 Masking Schemes*, IACR Cryptology ePrint Archive **2018** (2018), 492.

MS06. Florence Jessie MacWilliams and Neil James Alexander Sloane, *The Theory of Error-Correcting Codes*, 12 ed., North-Holland Mathematical Library, North-Holland, 2006.

RS85. Ron M. Roth and Gadiel Seroussi, *On generator matrices of MDS codes*, IEEE Trans. Information Theory **31** (1985), no. 6, 826–830.

Sag16. The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 7.4)*, 2016.

Sch80. Jacob T. Schwartz, *Fast Probabilistic Algorithms for Verification of Polynomial Identities*, J. ACM **27** (1980), no. 4, 701–717.

Wie99. Michael J. Wiener (ed.), *Advances in Cryptology — CRYPTO '99*, vol. 1666, Springer, 1999.